Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Andrea Schnell

Optimal Bluetooth Navigation Sensor Placement

Simulation-based Optimization and Evaluation in Outdoor and Indoor Scenarios

Master's thesis in Cybernetics and Robotics
Supervisor: Kristoffer Gryte
Co-supervisor: Torleiv H. Bryne
May 2023

NTNU
Norwegian University of
Science and Technology

Andrea Schnell

Optimal Bluetooth Navigation Sensor Placement

Simulation-based Optimization and Evaluation in Outdoor and Indoor Scenarios

Master's thesis in Cybernetics and Robotics
Supervisor: Kristoffer Gryte
Co-supervisor: Torleiv H. Bryne
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

MASTER'S THESIS DESCRIPTION SHEET

Name:	Andrea Schnell
Department:	Engineering Cybernetics
Thesis title (Norwegian):	Optimal plassering av Bluetooth-baserte sensorer for navigasjon.
Thesis title (English):	Optimal Bluetooth Navigation Sensor Placement
Thesis subtitle (English):	Simulation-based Optimization and Evaluation in Outdoor and Indoor Scenarios.

Thesis Description: Global navigation satellite systems (GNSS) is the primary outdoor positioning sensor for robotic vehicles such as unmanned aerial vehicles (UAVs). However, GNSS is susceptible to interference, natural and intentional, due to its low signal power. Another problem with GNSS is that it does not work indoors.

Low-cost Bluetooth sensors based on phased array radio systems technology have been shown to have potential in navigation systems. However, in contrast to GNSS, these sensors have limited coverage and are relative. The placement of such sensors is important to leverage the opportunities such sensor enables. Especially if there are strict requirements on navigation accuracy. Another aspect is economics. Even though Bluetooth technology is low-cost over-deployment should be avoided.
This thesis should cover two scenarios related to Bluetooth based navigation:

1. Navigation of a fixed-wing UAV during landing
2. Localization of robots in an indoor robotic warehouse

The following tasks should be considered:

1. Perform a short literature review on
    a. Bluetooth communication technology and its applicability for navigation and on
    b. Kalman filtering for navigation and the Cramér-Rao lower bound (CRLB) on estimation error.
2. Implement a simulator simulating relevant sensor measurements for UAV.
3. Implement an estimator for estimating the UAV's motion state and derive a method for placing the Bluetooth sensors limiting the estimation error based on the CRLB.
4. Develop an algorithm that can optimally place Bluetooth navigation sensors in the warehouses.
5. Present and discuss your results. Discuss potential tradeoffs w.r.t to accuracy and number of sensors in use.
6. Conclude on your results and suggest further work.

Start date:	2023-01-09
Due date:	2023-06-05
Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Reseacher Kristoffer Gryte, Dept. of Eng. Cybernetics, NTNU
Co-supervisor:	Associate professor Torleiv H. Bryne, Dept. of Eng. Cybernetics, NTNU

Preface

This master's thesis is carried out at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU) during the spring of 2023. The thesis is submitted in partial fulfillment of the requirements for the Master of Science degree. The idea of navigation of unmanned vehicles using Bluetooth was proposed by Kristoffer Gryte.

I would like to thank my supervisor, Kristoffer Gryte, for his support during the fall of 2022 on the specialization project and his great help during the first months of working on the master thesis. Then I would like to thank Torleiv Håland Bryne who was able to quickly familiarize himself with the research and provide valuable supervision during the final months of the thesis.

A big thanks goes to my boyfriend and family for their support and help with proofreading my thesis. Lastly, I want to thank my friends who have made my 5 years at NTNU unforgettable.

Andrea Schnell
Trondheim, May 2023

Abstract

Over the last few years, there has been an evolution in technology that has caused a demand for increasing abilities to navigate in various environments. The well-known, and widely used, Global Navigation Satellite Systems (GNSS) provide adequate navigation accuracy, but they are limited to navigation outdoors and with meter-level accuracy. New methods that can provide robust navigation with a sub-meter level accuracy indoors and outdoors are advantageous to explore.

This thesis explores the possibilities of utilizing various Bluetooth Low Energy (BLE) sensors for navigation. BLE has two ways of positioning: Direction Finding or beacons for distancing. The thesis demonstrates how Bluetooth sensors can be used to provide viable navigation solutions when GNSS are not available by simulating two scenarios: one situated outdoors and the other indoors. The outdoor scenario considers an Unmanned Aerial Vehicle (UAV) that needs navigation when flying toward a landing net. The navigation is performed by estimating the UAV's state with an Extended Kalman Filter (EKF), then the achievable precision is evaluated using the Cramér-Rao Lower Bound (CRLB) method. The other scenario mimics an autonomous warehouse scenario, requiring constant monitoring of moving robots. To classify the obtained coverage by the Bluetooth sensors, a maximum likelihood estimator is used to evaluate how well every point in the warehouse is covered. The CRLB is also applied in this case to assess the method.

Throughout the thesis, a more thorough understanding of how Bluetooth devices can be used for navigational purposes is provided, with a special focus on the optimization of sensor positions for ideal geometry to limit the number of sensors needed. Bluetooth devices are easy to deploy and low-cost. However, BLE communicates with radio frequency (RF) signals which makes them prone to errors and noises. However, the results provided in this thesis demonstrated how it is possible to obtain accurate navigation while remaining a low-cost system. The conclusion is that it is possible to implement a navigation system with Bluetooth devices with respect to the wanted use case. The necessary accuracy can be achieved if one is willing to invest in equipment.

Sammendrag

I løpet av de siste årene har det vært en utvikling innen teknologi som har forårsaket et behov for økende evne til å navigere i ulike miljøer. De velkjente og velbrukte Global Navigation Satellite Systems (GNSS) gir tilstrekkelig navigasjonsnøyaktighet i mange tilfeller, men de er begrenset til navigasjon utendørs og med nøyaktighet på meternivå. Nye metoder som har mulighet til å gi robust navigasjon med en nøyaktighet på under en meter, både innendørs og utendørs, er gunstig å utforske.

Denne oppgaven utforsker mulighetene for å bruke ulike Bluetooth-sensorer for navigasjon. Bluetooth Low Energy (BLE) har to måter oppnå posisjonsinformasjon på, enten ved å finne retningen (Direction Finding) eller distansen (beacon range). Avhandlingen har som mål å demonstrere hvordan Bluetooth-sensorer kan brukes til å gi robuste navigasjonsløsninger ved å utføre to analyser: ett utendørs og det andre innendørs. Utendørsscenarioet tar for seg en Unmanned Aerial Vehicle (UAV) som trenger navigering når den flyr mot et landingsnett. Navigasjonen utføres ved å estimere UAV-ens tilstand med et Extended Kalman Filter (EKF), deretter evalueres den mulige nøyaktigheten ved å bruke Cramér-Rao Lower Bound (CRLB) metoden. Det andre scenarioet forestiller et autonomt lager-scenario, som krever konstant overvåking av roboter i bevegelse. For å klassifisere den oppnådde dekningen av Bluetooth-sensorene, brukes en maksimal sannsynlighetsestimator for å evaluere hvor godt hvert punkt på lageret er dekket av sensorer. CRLB brukes også i dette tilfellet for å vurdere nøyaktigheten av metoden.

Målet med oppgaven er å gi en mer grundig forståelse av hvordan Bluetooth-enheter kan utnyttes til navigering. Fokuset er på å optimalisere sensorposisjonene for ideell geometri for å begrense antall sensorer som trengs. Bluetooth-enheter er enkle å bruke og rimelige, men kommunikasjonsmetoden er ved bruk av radiofrekvenssignaler (RF) som gjør dem utsatt for diverse feil og støy. Resultatene i denne oppgaven demonstrerte imidlertid hvordan det er mulig å oppnå nøyaktig navigasjon samtidig som det forblir et lavkostsystem. Konklusjonen er at det er mulig å implementere et navigasjonssystem med Bluetooth-enheter med hensyn til ønsket bruk. Nødvendig nøyaktighet kan oppnås dersom man er villig til å investere i utstyr.

Table of Contents

Chapter 1

Introduction

The evaluation of technology and trends over the past decades has led to an increasing desire to improve the ability to navigate in various environments. Navigation capability has become standard practice, and many applications require a sub-meter level of accuracy [1]. Global Navigation Satellite Systems (GNSS) are the leading outdoor navigation and localization standard today. With GNSS, a user can obtain globally precise positioning, navigation, and timing (PNT) at a low cost [2]. However, GNSS suffer from disadvantages caused by its nature - using satellite systems. The low signal power causes GNSS to be highly sensitive to intentional jamming and spoofing, resulting in a big issue for safety-critical systems. In addition, the receivers need to be in a line-of-sight to the satellites, which limits the ability for indoor navigation [3]. The drawbacks of GNSS motivate the need for other navigation techniques which can provide adequate navigation in scenarios where there are limitations with GNSS.

Radio frequency (RF) technology can be applied as an alternative approach to GNSS for navigation and localization. There are many RF technologies already explored, such as Phased-Array Radio System (PARS) [2], Radio Frequency Identification (RFID) [4], Wi-Fi [5], and Ultra-Wideband (UWB) [6]. Recently, there has been a growing interest in using Bluetooth as a location technology. Bluetooth utilizes a standardized technology that operates on the open radio spectrum and has so far been the most successful wireless technology [7]. Bluetooth is easy to deploy at low-cost, making it an attractive technology to explore and utilize further. It has been shown that both Wi-Fi and UWB technologies provide more accurate positioning than Bluetooth, but they are more expensive and have technical limitations [5]. Previous experiments with Bluetooth RF signals have focused on using Bluetooth Low Energy (BLE) beacons and calculation of the Radio Signal Strength Indicator (RSSI) for ranging solutions [8–11]. A more recent method based on phased-based ranging has also been tested for indoor localization [12]. Location information is acquired by employing a trilateration algorithm, requiring multiple ranges available. There are many shortcomings when using RSSI. The main limitation is propagation effects, making it difficult to obtain accurate location information [8].

The Bluetooth Special Interest Group (SIG) is constantly working on improving Bluetooth technology. In 2019 a new Bluetooth version was released; core specification 5.1. This version introduced a new feature called Direction Finding, enabling the use of Bluetooth devices for high-accuracy localization [13]. Multiple tests have been performed on utilizing Direction Finding for short-range indoor localization [14–16], resulting in sub-meter levels of accuracy. While most of the previous tests have been indoors, [17] and [18] also performed experiments with Bluetooth Direction Finding for outdoor navigation where longer ranges were proven feasible. Indoor localization has proven to be more error-prone than outdoor, as a result of signals being easier reflected and obstructed in closed environments.

It is necessary to perform additional testing of Bluetooth for navigational purposes. Previous experiments have confirmed that it is possible to obtain accurate localization information, but it is still not acknowledged as a robust navigation system. Bluetooth navigation is limited by a relatively short feasible range, bounding the operational area. With the high-accuracy Direction Finding feature, Bluetooth still suffers from RF disadvantages, such as reflection errors like multipath and blockage of the signal [17]. Hence, systems relying solely on Bluetooth navigation might not be sufficiently robust. The fusion of multiple Bluetooth sensors has the potential to generate a complete 3D position; however, it may be necessary to integrate additional metrics to ensure robust accuracy for navigation [19].

1.1 Main Contributions

The purpose of this thesis is to provide additional analysis of cases where using Bluetooth for navigation can be advantageous. A greater understanding of its potential use cases is achieved by demonstrating two different scenarios where the use of Bluetooth provides sufficient accuracy for navigation. Both traditional BLE beacons for range measurements and newer Bluetooth Direction Finding devices will be used. The simulation study investigates how sensor placement and the number of available sensors affect the accuracy of the systems. The systems can be optimized to achieve the best possible accuracy by leveraging knowledge about the uncertainties in Bluetooth measurements. This optimization is achieved by exploring the Cramér-Rao Lower Bound (CRLB) of the variances for the various cases [20]. The variances enable a determination of whether Bluetooth can be used for navigation while maintaining adequate precision.

The approach is to provide two distinct scenarios that require different functionality from the navigation system. One scenario is placed indoors where multiple Bluetooth sensors are needed to provide sufficient coverage in a warehouse. This case provides knowledge of how Bluetooth sensors can be used for indoor industrial navigation systems where good accuracy is critical. The other scenario is with an Unmanned Aerial Vehicle (UAV) flying outdoors. This requires sufficient navigation accuracy to properly land. There are sensors placed in the landing area, and some additional sensors are placed nearby to support the measurements. In

the UAV case, the noises present in the Bluetooth measurements are handled using the Extended Kalman Filter (EKF) as an estimator. A maximum likelihood estimator is utilized for the indoor positioning scenario. The goal for both scenarios is to optimize the sensor placements such that the CRLB is below a threshold to provide robust navigation and localization.

1.2 Organization

First, the theoretical foundation is described in Chapter 2, including limitations of GNSS and how Bluetooth can be used for navigation. Thereafter, relevant theory of EKF and CRLB calculations is included. Chapter 3 covers how the two scenarios will be simulated, and Chapter 4 provides the different mathematical models of the systems and how the estimation and optimization are performed. The obtained results are presented in Chapter 5 and then discussed in Chapter 6. The discussion evaluates the obtained results and possible limitations of the implemented methods. Lastly, Chapter 7 contains a conclusion of how using Bluetooth to navigate these cases works, including thoughts about further work.

Chapter 2

Background Material

2.1 Bluetooth

Bluetooth is said to be the most successful wireless technology in history [7]. It is wildly used today in everyday life. The technology has for a long time been the leader in audio space but is now experiencing increased usage for non-communication purposes. There are several benefits with Bluetooth: such as low-cost, low energy consumption, independent of networks, easy to deploy, open radio spectrum, and more. With these benefits and the broad use of Bluetooth, extending the Bluetooth functionality is a popular topic of research [19].

Bluetooth is a wireless communication technology that communicates using low-power radio waves. BLE is a subset of the Bluetooth protocol. It is designed for low power consumption and machine-to-machine communication. Bluetooth uses signals with frequencies between 2.40-2.41GHz. BLE divides the band into 40 channels, each with 2GHz width. The messages either contain data or advertisement. Advertisement messages are needed to enable communication and are therefore the chosen message type for positioning [21].

2.1.1 Limitations in Other Navigation Techniques

One of the most known navigation techniques is GNSS. A well-known system like this is the Global Positioning System (GPS). GNSS are available worldwide and can be used for navigation on land, sea, and in the air. Both civilians and high-level security services as the military are using GNSS navigation. It includes an end user which is equipped with low-cost, lightweight receivers that measure the GNSS data signals transmitted from various satellites. From these signals, the end user can derive information about its location with high accuracy. Stand-alone GNSS are typically able to provide user positioning with an accuracy of approximately 4m [22].

Due to its attractive features, GNSS are the primary positioning solution when working with UAVs [23]. However, there are issues with this navigation technique. One of the issues is to accurately determine the position of the satellite at

a given time. When calculating the end user's position, the system needs information about when the signal was transmitted and when it was received. This makes the results dependent on two clocks, one on the satellite and one on the receiver. As a result, there will be a clock bias in the range measurements. Therefore, the derived results are only pseudo-ranges [24]. Other errors in the pseudo-ranges are orbital errors, measurement noises, and different delay factors. There are methods for reducing these errors following the issue with GNSS, however, there will still be some localization errors present.

Other issues that can disturb the navigation with GNSS are caused by radio frequency interference (RFI). Due to GNSS's low signal-to-noise ratio (SNR), the systems are prone to both intentional and unintentional interference. Intentional interference such as jamming and spoofing makes the GNSS services vulnerable to malicious attacks that are designed to mislead the navigation. This type of interference may be critical for the navigation of a system. One way of trying to reduce these errors is to use extra navigation systems to support the GNSS [23]. Another vulnerability of the systems is that the GNSS signals are very weak when they reach the Earth's surface. This makes the signals prone to disturbance and blockage. Furthermore, it is not possible to use in tunnels or inside buildings as they cannot pierce through concrete [22]. Therefore, it is necessary to have other reliable navigation systems for these use cases.

Since GNSS are not usable for indoor positioning and localization, other methods must be applied. Indoor positioning has been a research topic for over two decades, where most of the focus has been on using Wi-Fi and RFID [25]. However, compared to the accuracy provided by GPS, these indoor position techniques do not provide a robust implementation in the consumer market. Using Wi-Fi for positioning can be beneficial as it is a widespread technology [26]. It can be provided through already existing infrastructure, but the access points (APs) are primarily designed for communication and not positioning. The APs may be owned by organizations that have the ability to modify or deactivate them, which creates greater unpredictability. Moreover, deploying a large number of APs for positioning purposes leads to significant expenses [21]. RFID systems can be active or passive and consist of three components; the reader, the antenna, and the tag. The positioning is based on using RSSI, Angle-of-Arrival (AoA), fingerprint, and more. It has the advantages of being lightweight, low-cost, and suited for asset tracking by using RFID tags. However, it requires additional infrastructure and is difficult to deploy for indoor positioning [25]. BLE has a relatively new positioning technology. It is of growing interest in studying BLE for indoor positioning methods due to its potential and advantages.

2.1.2 Bluetooth Direction Finding

In 2019, the Bluetooth Special Interest Group (SIG) presented Bluetooth Core Specification v5.1 [13]. This version introduced a new direction finding feature that made it possible for Bluetooth devices to determine the direction of a trans-

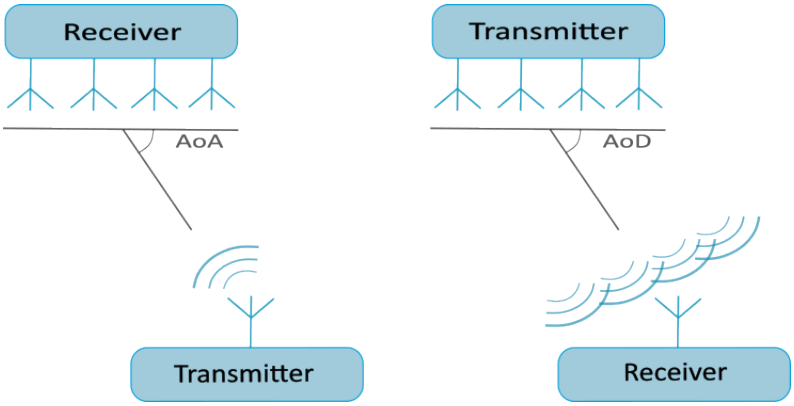


Figure 2.1: The left figure illustrates AoA where the antenna array is the receiver. The right figure illustrates AoD where the antenna array is the transmitter.

mitted Bluetooth signal with high accuracy. The BLE controller has been enhanced so that it contains specialized hardware with an antenna array. The array can be used in the calculation of the receiving signal's direction.

Bluetooth Direction Finding offers two different methods for positioning which use the same underlying basis, but the preferred method differs based on the situation. The methods are Angle-of-Arrival (AoA) and Angle-of-Departure (AoD). In both cases, the direction finding signals are transmitted by one device and then received by another device which uses the signal to calculate its direction [27].

In AoA, there is one moving tag which is the transmitter. A base station with multiple receivers is placed in an antenna array. The receiving part then calculates the direction from which the signal comes. AoA can be a preferred method in scenarios where there is one moving device and multiple receivers that want to measure the direction simultaneously. However, if there is more than one transmitting device, there may be issues with colliding signals. This results in erroneous signals reaching the antennas, hence calculating the wrong directions. In this case, AoD is the preferred method. In the AoD case, the antenna array is the transmitter, while the moving devices are the receivers. It is the moving device's job to calculate where the antenna is placed, and determine its position from this information [17]. The difference between AoA and AoD is shown in Figure 2.1.

The data used when calculating the direction is based on some of the fundamental properties of radio signals. The frequency of a radio signal varies depending on which channel the BLE device uses. The wavelength of a signal is depending on the frequency. Wavelength is a measure of the distance between the start and end of one wave cycle and is an important factor when calculating the signal's direction. Another value that explains a radio signal is its phase. The phase is measured as an angle from 0 degrees at the start of the wave cycle, increasing throughout the cycle, until it reaches 360 degrees at the end. The phase is measured as the angle of the signal as it passes through an antenna.

When a Bluetooth signal passes over an antenna, the phase at that moment is

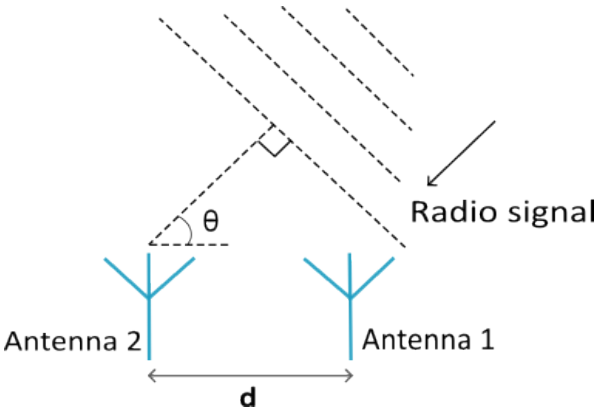


Figure 2.2: Calculation of Angle-of-Arrival, close up of two antennas receiving a signal. Figure taken from [28].

measured. If another antenna is placed in the same area as the first one, but at a slightly different distance from the transmitting device, the antenna will be able to measure the same Bluetooth signal. However, when measuring both phases at the exact same time, the measured phases in the two antennas are slightly different due to the different travel distances of the signal. With a known frequency of the signal, i.e., known wavelength, and known distance between the two antennas, it is possible to calculate the direction of the transmitted signal. A figure of the explained principle is shown in Figure 2.2. The angle of arrival θ can be calculated as

$$\theta = \arccos\left(\frac{\psi\lambda}{2\pi d}\right), \tag{2.1}$$

where λ is the wavelength, the measured phase difference is ψ, and d is the distance between the antennas [27].

The design that the antennas are placed in, as well as the number of antennas used in an antenna array, can vary. This makes it possible to decide the complexity of the calculations and which angles the array measures. Given a linear array where all the antennas are placed in one line results in simple calculations of a single angle. This is shown in the example Figure 2.2. A more complex design as a rectangular array is able to calculate two angles, for example both azimuth and elevation.

Bluetooth signals transmit information by using two frequencies that alternate between representing binary 1 and 0. One frequency corresponds to a 1, while the other corresponds to a 0. As the wavelength of a signal is dependent on its frequency, the wavelength changes with the alternating frequencies. Accurate phase measurements are crucial in the direction calculations, and these depend on having a constant wavelength. To overcome this issue, a Constant Tone Extension (CTE) is added to the end of the Bluetooth signals. The CTE consists of a series of symbols each representing binary 1. When a series of binary 1s are sent, the frequency and wavelength remain constant. This makes it possible for the antennas

in the array to correctly measure the phases [27].

In the direction calculation process, all antennas in the array measure the same Bluetooth signal. By utilizing the known distance between them and the emitted signal's wavelength, the direction can be calculated. However, the antennas do not measure the signal simultaneously. Instead, a switching technique is used to enable one antenna to measure at a time, thus keeping the hardware system at a low-cost. As a result, phase measurements are taken sequentially. This requires a more complex software system than if the measurements were taken simultaneously. The incoming signals are sampled by taking measurements at given intervals. The sampling process In-Phase and Quadrature (IQ) measures the phase and amplitude of signals in a logical sequence, allowing for the data to be compared and processed by each antenna. This makes it possible to determine the angle at which the signals arrive [27].

Sollie et al. [17] experimented on how the new Direction Finding feature can be utilized to estimate direction. In their study, they had an array consisting of 12 antennas and wanted to estimate the AoA of a fixed-wing UAV. The method of estimating the angles based on the IQ measurements consisted of four steps resulting in values for the azimuth and elevation angle. During the estimation, there were factors that needed to be accounted for. Some of the factors were the carrier frequency offset (CFO), far-field assumptions to obtain plane waves, and assumptions related to multipath. Multipath is when the signal has been reflected by a surface before it reaches the receiver. The reflection changes the direction of the signal, which results in wrong information regarding where the signal was sent from. Sollie et al. [17] concluded that they were able to sample measurements at a maximum rate of 10Hz. This was in order to stay within the link limit and be able to identify the measurements in time. They were able to use Direction Finding for outdoor navigation with a distance of up to 700m without a significant reduction in accuracy. The azimuth angle had the best accuracy with a standard deviation (std) of 1°. The elevation angle was harder to obtain accurately. The elevation suffered from multipath, which created unreliable and erroneous results. The main issue in outdoor navigation is multipath caused by the ground surface, resulting in large errors in the elevation direction. Different factors affect the elevation error, such as the height of the UAV over a surface, the signal frequency, the antenna pitch angle, and antenna size. It is to some extent possible to remove the multipath effect by calibration. However, as the factors are depending on the environment, this is not possible to implement in general [17].

2.1.3 Bluetooth Range Estimation

Direction Finding is a feature that recently got included in the BLE specification for supporting high-accuracy direction results. There is not yet released a feature in the Bluetooth specification for high-accuracy range estimation. However, as of May 2023, the Bluetooth SIG is working on adding a growing set of positioning capabilities to the technology, including high-accuracy range estimation. The

current positioning technology contains advertising for presence, RSSI for basic distancing, and Direction Finding. With a high-accuracy distancing feature, it is possible to create locating systems which provide an even higher level of accuracy than with the current technology [29].

The current distance measurements are performed with BLE beacons which were introduced in 2010. The beacons use RSSI methodology to calculate the range. The obtained positions from the calculations are based on received signal power or strength, which provides limited accuracy. The calculation of the distance is generally performed by using the logarithmic distance path-loss model. This model takes the measured RSSI, a path-loss parameter, and a Gaussian random variable, and calculates the assumed distance between a receiver and a transmitter. The limited accuracy from RSSI distancing comes from multiple factors, such as noise present in the signal, time-varying characteristics in the BLE signal, and environmental issues such as multipath [30]. Typical accuracy can vary from $2-10$m. This may be acceptable in multiple use cases, but not accurate enough to compete with new high-accuracy positioning systems [31]. However, there are benefits of selecting Bluetooth technology for location instead of other traditional radio frequency devices. The accuracy of beacon positioning can be improved by relying on multiple measurements and investigating different filtering techniques to reduce the fluctuations. The works of Sukmar [19] have looked into the details of how distancing with Bluetooth technology can be improved using various ranging techniques. It is shown that combining the RSSI distance with direction finding technologies improves the accuracy compared to RSSI alone. However, adding the additional capabilities results in increased complexity of the systems and raises the infrastructure cost. Other options for increasing the accuracy of Bluetooth ranging can be to measure the time of flight and the phase of the signal.

Experiments with phase-based ranging to provide high-accuracy distancing with BLE were performed by Zand et al. [12]. A phase-based solution uses the number of signal-phase shifts during the signal propagation between the transmitter and receiver. The method is able to reduce the multipath problem by utilizing two devices that measure the phase changes over multiple frequencies. The procedure of using multi-frequencies for phase-based ranging is called Multi-Carrier Phase Difference (MCPD). MCPD can be used for ranging with BLE by utilizing the channels for hopping between tones in the entire BLE frequency band. The conclusion formed by Zand et al. [12] was that different factors would affect the ranging error. However, many of the parameters in the system are known beforehand, making the ranging error predictable.

2.1.4 Limitations with Bluetooth

The use of Bluetooth signals for direction and range estimation is subject to several limitations that may hinder its use for navigation purposes. Unlike GNSS, navigation with BLE does not have global coverage. The navigation only works in

predefined areas where the sensors are placed in suitable positions with respect to what they are measuring. This results in a limited area where the navigation is accessible. The need for relative positioning limits the range and accuracy of the possible operational area for BLE navigation. Direction Finding devices have been shown to have a feasible range of up to 700 meters [17]. On the other hand, the range measuring beacons appears to be limited by a feasible range of only 125 meters. This ranging result was obtained by a colleague Hanne Loftsnes. Her work concluded that MCPD gives the best range estimates. If navigation is only performed using one measurement for range and one for direction, the final feasible distance to obtain a 3D position is then limited by the measurable range. This is because both measurements are needed for a full position calculation.

The limited area is known to the user beforehand. Consequently, the user is able to take it into account when designing the navigation system. In navigation scenarios where BLE navigation is considered, there are cases where there is a need for local navigation within a specified area. Hence, the impact of that issue is reduced. Nonetheless, there are other issues that need to be taken into account. For instance, the fact that the sensors are measuring a signal traveling from one part to another. This makes the signal vulnerable to errors such as obstruction and reflections. An obstruction is when something is in the way and blocking the signal from reaching the receiving antenna. In the case of an obstruction, the sensors are not able to measure the signal and remain clueless. If a traveling signal hits an object, the signal may be reflected, and its direction changes. If a reflected signal reached a receiving antenna, the signal has traveled further than the original distance and the angle has been modified. This results in the receiving antenna calculating a totally wrong position of where the signal was sent from.

How much the issues of obstruction and reflection are affecting Bluetooth Direction Finding were tested by Sambu et al. [18]. They performed experiments both indoors and outdoors in order to see how much obstructions and reflections by the environment affected the accuracy. Their results showed that the error was smaller when measured in an outdoor environment than indoor. The angle error outdoor was on average 0.48°, with a standard deviation of 0.28°. On the other hand, the error indoor averaged at 1.83°, with a standard deviation of 1.1°. From these results, the angular error for outdoor environments was measured as 73% smaller than for indoor environments. Sambu et al. [18] also performed positioning experiments, where the results for outdoor positioning had an average error of 22cm while the indoor average was 36.5cm.

According to a technical overview by SIG member Woolley [27], the Bluetooth Direction Finding feature can be utilized to develop positioning systems that can operate with sub-meter accuracy. The paper mentions examples of indoor use cases such as positioning and asset tracking. The inaccuracy in the range sensors varies more than the direction sensors. The error is dependent on the type of ranging method used, the distance between the two points, and the utilized algorithm. The results Zand et al. [12] obtained indicated a ranging error of approximately 1 meter with one method, but another method indicated an average

error of up to 5 meters.

2.1.5 Combination of Bluetooth Measuring Devices

A BLE device is only able to either measure the direction of a transmitter or the range between two devices. Given one of these measurements, it can only be deducted that the transmitting device is somewhere on a circle with a specified radius or on the line of direction. As a result, in order to determine a position using these sensors, it is necessary to have access to measurements coming from multiple sensors.

A position can be determined from range measurements given that the sensor positions are known. If a sensor measures the distance to a point, it can only determine that the point is situated on a circle with a radius equal to the measured distance from the sensor. If two distance measuring sensors are used, their respective circles will intersect at two points. Therefore, the target could potentially be located at either intersection. However, adding one more distance sensor to the scene allows for the identification of a single intersection point where all three circles intersect. The intersection then represents the location of the target. The technique is called trilateration, where a minimum of 3 range measurements are required to determine a position [9]. The position of the target is then calculated by solving a set of equations based on the measured ranges and corresponding sensor position. For n measuring sensors in a 2D case, the set of equations will be

$$\begin{aligned} d_1 &= \sqrt{(x_u - x_1)^2 + (y_u - y_1)^2} \\ &\vdots \\ d_n &= \sqrt{(x_u - x_n)^2 + (y_u - y_n)^2}, \end{aligned} \tag{2.2}$$

where (x_u, y_u) is the target's position and $d_\bullet$ is the relative distance. An issue with this positioning technique is the uncertainties in the distance measurements. BLE range measurements can contain a great amount of errors. Hence, when calculating the position of the target, the result will depend on these errors. The intersection points of the ranges can be explained as an area. The size of the area is decided by the assumed errors as well as the geometry of the sensors. The positioning of the sensors with respect to each other plays a role in the accuracy of the calculated position. This geometric influence is known as Dilution of Precision (DOP), and represents how much the accuracy of the solution is diluted due to the sensor-target geometry [32]. How trilateration works are visualized in Figure 2.3. The figure shows the true distances as $d_\bullet$, while the striped lines illustrate the possible measured ranges due to the errors. The red dot is the actual position, while the red area illustrates where the position may be deduced to be due to the errors. Good geometric diversity of the sensors reduces the size of the error area, hence improving the accuracy of the position estimation [22].

One measure to improve the trilateration results is to over-determine the system. This is done by using more sensors than required to determine the intersec-

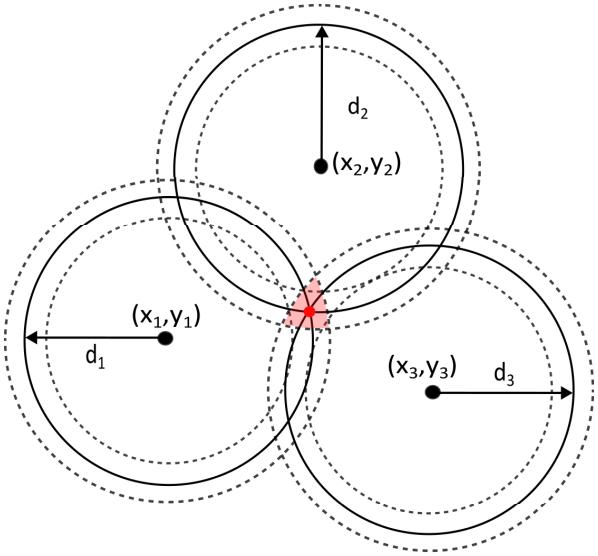


Figure 2.3: Illustration of trilateration using three sensors. The red dot is the actual position, while the colored area is possible placements due to errors in range measurements.

tion of the circles. This gives a greater number of equations than the number of unknowns and reduces the impact of the error.

For Direction Finding devices, a single sensor only knows where on a line the target is positioned based on the calculated elevation and azimuth. With two sensors measuring the direction of the same target point, the location of that target can be derived using the two measured angles and the known positions of the two sensors. This is illustrated in Figure 2.4. There is also some error present in the direction finding techniques, which may result in uncertainty in the localization of a target. Improvement can be made by using extra measuring devices to reduce the impact of errors.

Using a combination of a direction finding Bluetooth device and a range-estimating beacon also results in position estimation. However, since range estimation results in relatively crude distance estimation, the final location calculation may not be very accurate. It can be improved by using the Direction Finding device together with multiple beacons. This benefits from the accuracy of Bluetooth Direction Finding and limits the range error by trilateration [27].

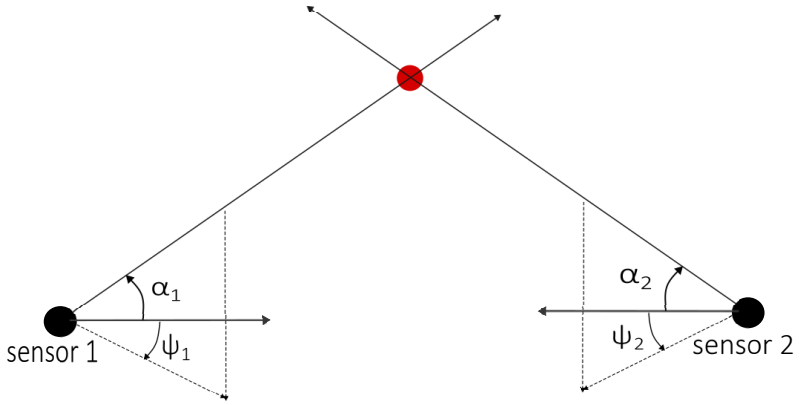


Figure 2.4: Illustration of target localization using two direction finding devices. Target is the red dot and the direction lines pass through it.

2.2 Inertial Navigation System

An Inertial Navigation System (INS) consists of an Inertial Measurement Unit (IMU) and numerically integrated kinematic equations. An IMU typically contains a three-axis accelerometer and a three-axis rate gyro. Sometimes it also contains a three-axis magnetometer. The INS is able to estimate the position, velocity, and attitude of a device it is mounted on. The position is obtained by integrating the acceleration measurements twice while the gyro measurements are integrated once to obtain the orientation. The kinematic equations expressing the position and velocity states

$$\dot{\mathbf{p}}^n_{m/n} = \mathbf{v}^n_{m/n} \tag{2.3}$$

$$\dot{\mathbf{v}}^n_{m/n} = \mathbf{R}^n_b(\boldsymbol{\Theta})(\mathbf{a}^b_{imu} - \mathbf{b}^b_{acc}) + \mathbf{g}^n \tag{2.4}$$

where $\mathbf{a}^b_{imu}$ and $\boldsymbol{\Theta}$ is the measured acceleration and rotation respectively. $\{b\}$ is the body frame of the moving device, $\{m\}$ is the local INS measurement frame and $\{n\}$ is the North-East-Down (NED) frame. $\mathbf{R}^b_n(\boldsymbol{\Theta})$ is the rotation matrix from $\{b\}$ to $\{n\}$. $\mathbf{b}^b_{acc}$ is the accelerometer biases and $\mathbf{g}^n$ is the gravitation constant, $\mathbf{g}^n \approx [0, 0, 9.81\text{m/s}^2]^\top$. It is possible to include a state to estimate the bias as well [33].

If the measurements from an IMU were perfect, a stand-alone INS would be able to determine the exact position and orientation of a moving device relative to its initial state. However, this is not the case in reality as navigation depending solely on the INS leads to a drift in the computed outputs due to biases that exist in the sensors. The sensors can be calibrated before usage to remove the run-to-run internal biases. However, in-run biases change during navigation may be caused by misalignments and temperature changes affecting the output. In addition, there are some noises affecting the measurements, even though they generally are low. As a result, an INS is not adequate for navigation by itself, but may have beneficial effects when combined with another navigation system [33].

The gyroscope and accelerometer are the key components of an IMU. The classic gyroscope consists of a spinning wheel to measure the angular velocity. The wheel rotates about an axis and belongs to a gimballed system. This utilizes the conservation of momentum to measure the rotation. An accelerometer measures the result of forces acting on a body, and different principles can be used. A mechanical accelerometer is the simplest form and can consist of a pendulum. It measures acceleration by using Newton's second law of motion, $F = ma$. a is the acceleration caused by the force F acting on the body with a mass m. A vibratory accelerometer measures the frequency shifts resulting from the tension of a string being decreased or increased. The acceleration is proportional to the change in frequency [33].

The measurement equations for the acceleration and the angular velocity from the IMU is

$$\mathbf{a}^b_{imu} = \mathbf{R}^b_n(\boldsymbol{\Theta})(\dot{\mathbf{v}}^n_{m/n} - \mathbf{g}^n) + \mathbf{b}^b_{acc} + \boldsymbol{w}^{\,b}_{acc} \tag{2.5}$$

and

$$\boldsymbol{\omega}^b_{imu} = \boldsymbol{\omega}^b_{m/n} + \mathbf{b}^b_{gyro} + \boldsymbol{w}^{\,b}_{gyro} \tag{2.6}$$

respectively. The frames are the same as expressed above. $\mathbf{b}^b_{acc}$ and $\mathbf{b}^b_{gyro}$ are biases in the accelerometer and gyro, respectively. $\boldsymbol{w}^{\,b}_{acc}$ and $\boldsymbol{w}^{\,b}_{gyro}$ are the zero-mean noises added to the sensors measurements [33].

The use of an INS has several advantages. It does not rely on external signals as it is completely self-contained, making it unexposed to external factors. An IMU operates at a high frequency, and the obtained measurements typically have low levels of noise. This results in smooth position and attitude outputs. These benefits are great when an INS is integrated with another navigation system which may suffer from loss of measurements or receives the measurements at a slow rate. In these cases, the other navigation system can handle the drift caused by the IMU by compensating with its own measurements, while the INS can be used to estimate the position and attitude in between the other system's measurements. This is a method where the two navigation systems are of complementary nature where the benefits of one compensate for the drawback of the other [34]. Including an INS in such systems also works as a security plan in case the other system is disturbed or blocked.

2.3 Estimators

Estimators are mathematical functions that use data to estimate the value of an unknown parameter or variable [35]. In navigation problems, estimators are often used to estimate the state of a system using measurements from various sensors. An example of such an estimator is the EKF. The precision and accuracy of an estimator can be evaluated using statistical metrics such as bias, variance, and Mean Squared Error (MSE) [36].

Given an unknown quantity x, an estimator is deriving information about x from some data z which is related to x. The relationship between x and z can be described by the probability density function (PDF) $f(x|z)$, which is the probability of x given data z. The likelihood of an estimator is the probability of observing the data z given that x occurred, $\mathcal{L}(z|x)$.

An estimator is a function that tries to guess a concrete value of what x is given the data z. If $x \in \chi$ and $z \in \mathcal{Z}$, the estimator is a function $f : \chi \rightarrow \mathcal{Z}$, such that $\hat{x} = f(z)$. $\hat{x}$ denotes the estimate of x [35].

The Bayesian estimator is an example of an estimator following the criteria explained above. In this approach, the prior knowledge of x is given as a probabilistic model $p(x)$ as well as the relationship $p(z|x)$. A Bayesian estimator can be either maximizing or averaging. In the maximizing scenario, the aim is to find the best value of x given z. Averaging estimators return a value of x which best represents the knowledge of x given z [35]. The EKF is an example of a Bayesian estimator [37].

The observed data that is used to estimate x will not perfectly resemble the true state. Different types of errors can occur during data collection such as measurement errors. Since the estimate is a guess based on the observed data, the estimated value differs from the true value. This results in an estimation error $\tilde{x} = x - \hat{x}$. The evaluation of an estimator's accuracy is done by considering its bias, variance, and MSE. The bias explains the expected error of the estimation, denoted as $E[\tilde{x}]$. An unbiased estimator has the expected error $E[\tilde{x}] = 0$. The variance explains how much the estimator spreads its estimates. MSE is a combination of both bias and variance, providing an explanation of the average squared difference between the estimates and the true value [35].

The variance expresses how precise the estimator is, i.e., the degree of repeatability. This tells how consistent the results from measurements are if repeated many times. Precision is not related to how close to a true value the results are. Accuracy on the other hand relates to how well the measurements indicate the true value. With good accuracy, the expected value is close to the true value. The accuracy is typically indicated by an estimator's bias [38].

2.4 Extended Kalman Filter

A Kalman Filter (KF) is a recursive filter that is able to take a series of noisy measurements and estimate the state of a dynamic system. The filter works for both linear and nonlinear systems. The use of KF is popular in navigation systems since it is able to remove both white and colored noise in the measurements from the estimated states. Furthermore, it can reconstruct states in case of missing measurements. A KF is divided into two steps; one for prediction and one for correction. The prediction step is only dependent on previous state estimates and possible internal measurements. The prediction is able to work even during loss of measurements or external sensor failure. In these cases, the filter acts as a predictor and the estimated states are based on the filter's knowledge of the system's

state. The prediction rarely works perfectly, causing a drift from the true states if not corrected for a long time. The corrector step of the KF executes as soon as measurements are available. It considers the estimated states and the incoming measurements and corrects the predictions in order to minimize the variance of the state estimates [33].

2.4.1 Discrete-Time Extended Kalman Filter

If the system to be estimated is nonlinear, an extension of the linear KF must be applied. This is known as the Extended KF (EKF). The EKF is capable of handling nonlinearities in the process model and the measurement model with respect to the process [39].

EKF is still able to use linear estimation techniques by handling the nonlinearities in the system by linearization. As a result of this, the EKF is mostly used when the nonlinearities are mild and easy to linearize. Different linearization techniques can be utilized. One of the techniques is to linearize around an operating point where the operating point needs to be decided. The linearization in EKF is taken around the estimated state from the previous step at each iteration [35].

In order to filter and estimate states using incoming measurements from different sensors, the EKF must be in a discrete-time format. This is due to the fact that all measurements are obtained through sampling and therefore sent to the filter at discrete time intervals.

Derivation of the EKF Equations

The EKF is a result of the Bayes filter after constraining the belief to be Gaussian, assuming the noise to be Gaussian, and performing linearization of the models. The constrained belief function considers the probability of having a state $\mathbf{x}_k$ given the prior belief and the measurements up to and including this time step. This is the same as the PDF representing the likelihood of $\mathbf{x}_k$. The Gaussian belief can be expressed as shown in (2.7). The left-hand side of the equation expresses the likelihood of the state $\mathbf{x}_k$, and the right-hand side shows that the likelihood is Gaussian distributed with mean $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_k$ [37]. $\hat{\mathbf{x}}_k$ is the a posteriori estimated state based on the a priori estimated state $\check{\mathbf{x}}_k$. A priori refers to the estimation performed before new measurements are included. $\check{\mathbf{x}}_k$ is solely based on the previous estimate of $\mathbf{x}$ and current control inputs and calculated using the process model equations. The a posteriori estimate is the estimated state computed from both the a priori state and new measurements and is found as the mean of the probability distribution.

$$p\left(\mathbf{x}_k \mid \check{\mathbf{x}}_0, \mathbf{u}_{1:k-1}, \mathbf{y}_{1:k-1}\right) \sim \mathcal{N}(\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k), \tag{2.7}$$

A nonlinear system with additive noise can be expressed on the discrete-time form

$$\dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{2.8}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k, \tag{2.9}$$

where k expresses the current time step, and $k-1$ refers to the states at the previous time step. The additive noises

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{2.10}$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{2.11}$$

are assumed to be Gaussian with zero mean and covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively.

The functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear and therefore need to be linearized around the mean of the current state estimate. The linearized functions are created from the Taylor expansion and can be written on the form

$$\boldsymbol{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) \approx \check{\mathbf{x}}_{k-1} + \mathbf{F}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)\tilde{\mathbf{x}}_{k-1} \tag{2.12}$$

$$\mathbf{g}(\mathbf{x}_k) \approx \check{\mathbf{y}}_k + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k|k-1}\right)\tilde{\mathbf{x}}_k, \tag{2.13}$$

where $\tilde{\mathbf{x}}_\bullet = \mathbf{x}_\bullet - \hat{\mathbf{x}}_\bullet$ is the error variables, and $\mathbf{F}_{k-1}$ and $\mathbf{G}_k$ are the Jacobians of the nonlinear functions. The calculations of $\mathbf{F}_{k-1}$ and $\mathbf{G}_k$ are performed by taking the Jacobians of $\mathbf{f}$ and $\mathbf{g}$ with respect to $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ [35]:

$$\mathbf{F}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) = \frac{\partial}{\partial \mathbf{x}_{k-1}} \boldsymbol{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)\bigg|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1}} \tag{2.14}$$

$$\mathbf{G}_k\left(\hat{\mathbf{x}}_{k|k-1}\right) = \frac{\partial}{\partial \mathbf{x}_k} \mathbf{h}(\mathbf{x}_k)\bigg|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}} \tag{2.15}$$

The calculation of the a priori state and measurement estimations can be written on the form

$$\check{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \tag{2.16}$$

and

$$\check{\mathbf{y}}_k = \mathbf{g}(\hat{\mathbf{x}}_k) \tag{2.17}$$

respectively.

By analyzing the current state $\mathbf{x}_k$ given the previous state $\mathbf{x}_{k-1}$ and the current control input $\mathbf{u}_k$, the statistical properties yields

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \approx \mathcal{N}(\check{\mathbf{x}}_k + \mathbf{F}_{k-1}\tilde{\mathbf{x}}_{k-1}, \mathbf{Q}_k). \tag{2.18}$$

The statistical properties for the current measurement given the current state can be expressed as

$$p(\mathbf{y}_k \mid \mathbf{x}_k) \approx \mathcal{N}(\check{\mathbf{y}}_k + \mathbf{G}_k\tilde{\mathbf{x}}_k, \mathbf{R}_k). \tag{2.19}$$

These two PDFs can be used to arrive at the final Gaussian PDF expression for the state estimation as explained in (2.7). The resulting PDF is shown in (2.20) [37]. The Kalman gain is expressed as $\mathbf{K}_k$ and is used as a weighting factor in the filter. The matrix $\mathbf{K}_k$ is multiplied with the innovation, $\mathbf{y}_k - \check{\mathbf{y}}_k$, and controls how much the recent measurements are affecting the state estimation. With a high Kalman gain, the measurements are heavily weighted, resulting in the state tightly following the new measurements. A small gain can imply that the measurements are noisy and erroneous, resulting in more trust in the model predictions [39].

$$\mathcal{N}(\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k) = \mathcal{N}\left(\check{\mathbf{x}} + \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k), (\mathbf{I} - \mathbf{K}_k\mathbf{G}_k)(\mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^\top + \mathbf{Q}_k)\right), \tag{2.20}$$

With all the above equations derived, the recursive EKF algorithm can be implemented. The steps in the algorithm are described in Algorithm 1.

Algorithm 1 Discrete-time extended Kalman Filter

Initialize:
$\mathbf{x}_0 = \mathbf{x}(0)$
$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^\top]$
Design matrices $\mathbf{Q}_k = \mathbf{Q}_k^\top > 0$, $\mathbf{R}_k = \mathbf{R}_k^\top > 0$
for k = 1 to N **do**
  Prediction step:
  $\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k$
  $\check{\mathbf{P}}_k = \frac{1}{2}(\check{\mathbf{P}}_k + \check{\mathbf{P}}_k^\top)$ ▷ Ensures $\check{\mathbf{P}}_k = \check{\mathbf{P}}_k^\top$
  $\check{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$

  Kalman Gain:
  $\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{G}_k^T\left(\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^T + \mathbf{R}_k\right)^{-1}$

  Correction step:
  if Available measurements $\mathbf{y}_k$ **then**
    $\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{G}_k)\check{\mathbf{P}}_k(\mathbf{I} - \mathbf{K}_k\mathbf{G}_k)^\top + \mathbf{K}\mathbf{R}\mathbf{K}^\top$
    $\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{g}(\check{\mathbf{x}}_k))$
  end if
  return $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{P}}_k$
end for

2.5 Cramér-Rao Lower Bound

Cramér-Rao Lower Bound (CRLB) is a lower limit of the variance of an unbiased estimator. The bound explains the best possible precision the given estimator

is able to achieve. With CRLB it is possible to, among other things, determine whether it is achievable to improve the performance if a more advanced estimator is used, if using time to tune the filter will be worth it, and if it is theoretically possible to achieve some accuracy requirements with this estimator [40].

2.5.1 Fisher Information

The Fisher information is a measure of how much information it is possible to extract about the parameter of the random variable's PDF [41]. Let a random variable z follow a conditional PDF $f(z|x)$, where x is the parameter of the distribution. Assuming that the PDF has twice continuous partial derivatives, the Fisher information can be calculated by the negative expected value of the twice partial derivative with respect to x of the log-likelihood function [40]. The reason for using the log-likelihood is that maximizing the likelihood is equivalent to taking the max of the log-likelihood, except the log-likelihood function is more convenient to work with. The Fisher information measures the curvature of the log-likelihood. The curvature is related to the accuracy of $\hat{\mathbf{x}}$: a low curvature says that the likelihood surface is flat near its maximum which makes it difficult to obtain the optimal parameter, versus a strong curvature which comes from a strong peak and a clearly defined maximum [42]. The Fisher information can then be written on the form

$$\begin{aligned}\mathcal{I}(x) &= -E\left[\frac{\partial^2 \ln p(z \mid x)}{\partial x^2}\right]\Bigg|_{x=x_0} = E\left[\left(\frac{\partial \ln p(z \mid x)}{\partial x}\right)^2\right]\Bigg|_{x=x_0} \\ &= E\left[H(x)\right]|_{x=x_0},\end{aligned} \tag{2.21}$$

where the $p(z|x)$ is the likelihood and x_0 is the true value of x [43]. The matrix $H(x)$ is the Hessian of the log-likelihood with respect to the parameter x. If the parameters are of multiple dimensions, the Fisher information is often specified as the Fisher information matrix (FIM), calculated as shown in (2.22). The operator $\nabla_{\mathbf{x}}$ denotes the partial derivative with respect to $\mathbf{x}$, and $\Delta_{\mathbf{x}}^{\mathbf{x}} = \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\top}$ is the Hessian.

$$\begin{aligned}\mathcal{I}(\mathbf{x}) &= -E\left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln p(\mathbf{z} \mid \mathbf{x})\right]\Big|_{\mathbf{x}=\mathbf{x}_0} \\ &= E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{z} \mid \mathbf{x}))(\nabla_{\mathbf{x}} \ln p(\mathbf{z} \mid \mathbf{x}))^{\top}\right]\Big|_{\mathbf{x}=\mathbf{x}_0}\end{aligned} \tag{2.22}$$

In this thesis, it is assumed that the parameter to be estimated is multivariate, i.e., $\mathbf{x} \in \mathbb{R}^{n \times 1}$. Hence, the FIM is used further in this thesis.

2.5.2 Cramér-Rao Lower Bound Calculation

The CRLB is an inequality that is calculated by taking the variance of any unbiased estimator $\hat{\mathbf{x}}$ bounded from below the inverse of the FIM. This can be expressed as

$$\operatorname{cov}(\hat{\mathbf{x}}(\mathbf{z})) \;=\; E\left[(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})^{\top}\right] \geq \mathcal{I}(\mathbf{x})^{-1}, \tag{2.23}$$

where $\mathcal{I}(\mathbf{x})^{-1}$ is the CRLB, expressing the lower bound of the estimator's variance. The CRLB explains the best possible variance an estimator is able to achieve given its probability distribution. If an estimator's variance is equal to $\mathcal{I}(\mathbf{x})^{-1}$, the estimator is called efficient [43].

Due to the twice partial derivation in the equation for the CRLB, the calculation may be complex. However, there are some cases that lead to simplifications of the CRLB calculation. One of these is the parametric CRLB. Given a predetermined trajectory, it is possible to calculate the CRLB for each point of that trajectory expressed as

$$\mathbf{P}_{k|k}^{CRLB}(\mathbf{x}_{1:k}). \tag{2.24}$$

In this case, the calculation of (2.24) equals the Riccati equation on the same form as the one for calculating the error covariance matrix $\hat{\mathbf{P}}$ in the EKF [40]. As the CRLB is an evaluation of an estimator given the perfect estimates, the CRLB of a predetermined trajectory can be calculated by using the known states as inputs instead of the previously estimated state. This results in the same algorithm as EKF, Algorithm 1, the only difference being that all the expressions are evaluated based on the true trajectory instead of the previous predicted state [44].

Chapter 3

Simulator Design

This thesis examines the feasibility of utilizing Bluetooth signals as a navigation system in a variety of environments. To achieve this objective, two distinct analyses will be performed to evaluate the robustness of Bluetooth-based navigation. The objective is to determine whether Bluetooth navigation can replace traditional navigation techniques in situations where the latter may not be appropriate or effective.

The two cases being tested involve the navigation of a UAV in an outdoor environment and the navigation of robots in an indoor warehouse setting. As neither of the scenarios has been tested in real-life, simulations will be used to replicate expected real-life behavior.

3.1 Net-Landing Scenario of a Fixed-Wing UAV

In the net-landing case, a fixed-wing UAV is expected to follow a trajectory that starts from far away and moves closer to a net that can catch the UAV. In this case, the UAV does not need a runway to land, and since the only requirement is that the UAV lands within the net, it is the size of the net that decides the required accuracy. This is a case that may be relevant if the UAV is used to explore something on the ocean, and needs to land on a boat. A video illustrating a net-landing experiment similar to this scenario is shown in [45].

3.1.1 Trajectory Simulation

A simulation of the scenario is necessary to create measurements that can be used to test how the system works with BLE sensors. The simulation is performed using the Matlab UAV Toolbox and its functionality for data processing and scenario simulation. The function `minjerkpolytraj` is used to create a realistic trajectory. This function takes in a set of input waypoints and their corresponding time points and generates a minimum jerk polynomial trajectory based on the inputs. By enabling `TimeAllocation` to true, the function optimizes a combination of jerk and total segment time cost, hence treating the input time points only as an initial guess for

the arrival at each time point. The outputs are positions, velocities, accelerations, and jerks at the given number of samples specified by `numSamples` and a new set of time points that corresponds to the new output vectors [46]. The resulting positions, velocities, and time points can then be used as inputs to the function `waypointTrajectory`. This Matlab function takes in a set of waypoints, the corresponding velocities, and the time of arrivals and generates a trajectory based on these values. From the output trajectory, it is possible to extract position, velocity, and acceleration expressed in NED frame [47].

The simulation is created with respect to one origin, which is placed in the antenna array. Since the simulation results in positions in the NED frame, it is possible to use the Matlab function `ned2aer` to express the azimuth, elevation, and range values measured from the array. These values are then divided into two lists, one containing the angles and another with the ranges to represent the different measurements. There are also assumed measurements coming from beacons somewhere in the simulation space. The ranges from the beacons are also calculated by `ned2aer`, only moving the origin from the array to the beacon positions, and then extracting the range.

3.1.2 Simulation Results

The sample rate used to create the measurements was set to 250Hz. This is the rate at which the IMU is assumed to provide the acceleration measurements [2]. This equals new measurements every 4ms. The BLE measurements are limited to an update rate of 10Hz, resulting in new measurements every 0.1s [17]. This means that the accelerometer measures 25 times for every incoming BLE measurement. To deal with the difference in sample rate in the simulated measurements, only every 25th measurement was considered for the direction and range sensors.

The chosen waypoints were created so that the trajectory would mimic a realistic trajectory of a UAV. The UAV performs a big turn before it tries to navigate straight toward the landing net. The trajectory is shown in Figure 3.1 and the accelerations are shown in Figure 3.2. The position values from the trajectory are the ones that will be used for angle and range calculations, while the accelerations are needed to mimic the IMU measurements.

Additional results from the UAV flight simulation are shown in Figure 3.3 and the trajectory shown from above is illustrated in Figure 3.4.

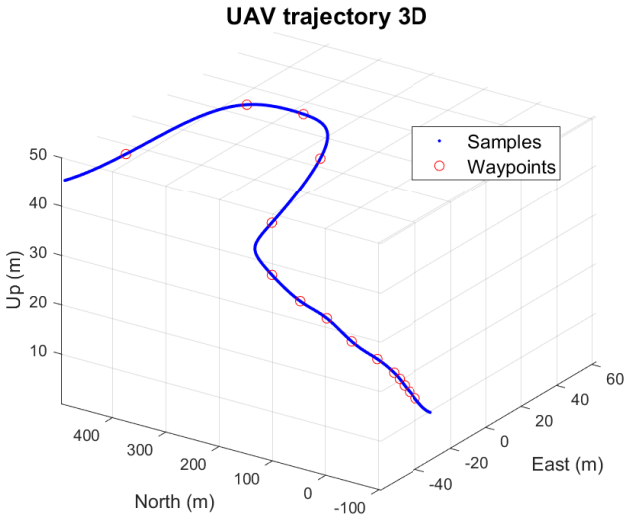


Figure 3.1: 3D plot of planned trajectory for the UAV flying towards the landing net in [0; 0; 0]. The red circles represent the chosen waypoints.

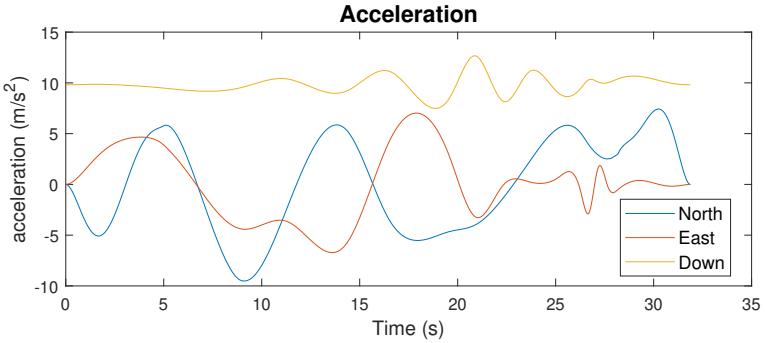
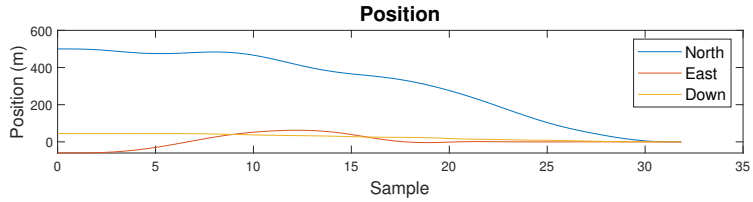
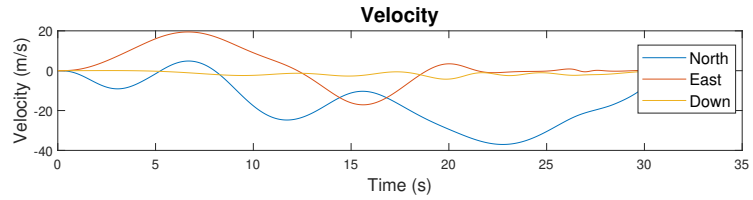


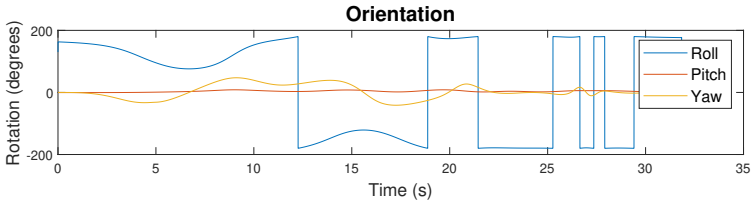
Figure 3.2: The acceleration of the UAV throughout the trajectory.



(a) Position in NED



(b) Velocity in NED



(c) Orientation in NED

Figure 3.3: Results from simulation of UAV trajecotry

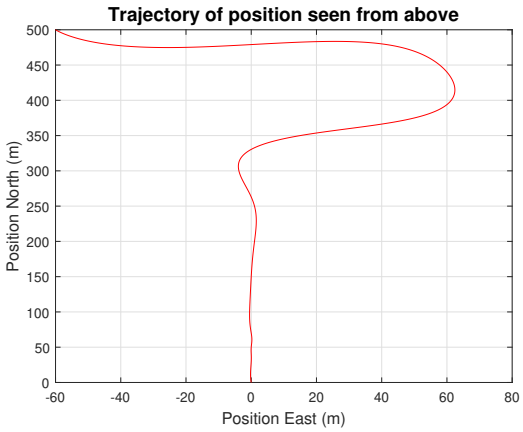


Figure 3.4: Trajectory of the UAV as seen from above.

3.1.3 Creating Erroneous Measurements

The simulation resulted in perfect measurements, making it necessary to define noises that were added to the measurements to create a realistic scenario. From the theory on Bluetooth sensors and their errors, Section 2.1.4, the white noises added to the direction measurements have a standard deviation of $\boldsymbol{\sigma}_\alpha = \boldsymbol{\sigma}_\psi = 0.28°$ while the range measurements have noises with standard deviation $\boldsymbol{\sigma}_\rho = 2\text{m}$ added. Additionally, it needed to be added both noises and biases to the acceleration measurements. The chosen values were taken from Gryte et al. [2], where the acceleration variances were set to $\boldsymbol{\sigma}^2_{acc} = 6.588 \times 10^{-4}\text{m}^2/\text{s}^3$, and the biases were set to $\mathbf{b}_{acc} = 50\text{mg} = 50 \cdot 9.81/1000\text{m/s}^2$ with variances $\boldsymbol{\sigma}^2_{b_{acc}} = 4.150 \times 10^{-6}\text{m}^2/\text{s}^3$. The biases were set to the maximum bias expected in an accelerometer. However, in reality, they can be both smaller and of negative value. The noisy azimuth and elevation measured by the antenna array are shown in Figure 3.5. Figure 3.6 shows the acceleration measurements where both noise and bias were added.

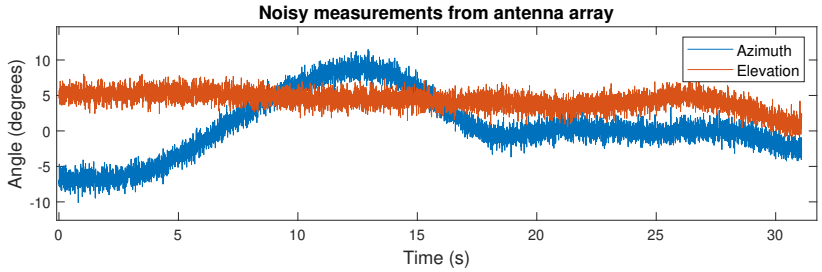


Figure 3.5: Direction measurements with added white noise.

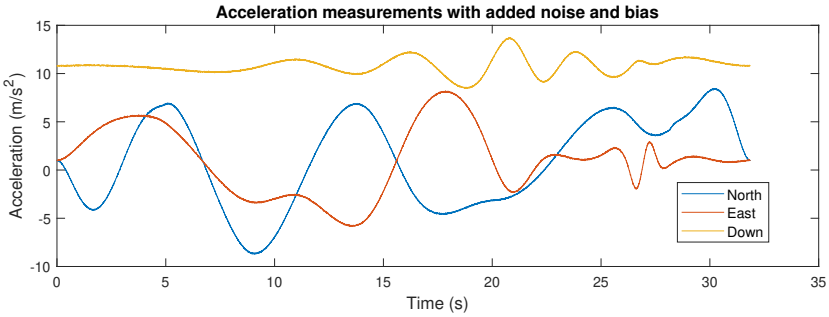


Figure 3.6: Acceleration measured by the IMU. Noise and bias are present.

3.1.4 Separate Measurements

Due to the sensors being placed at different points in the simulated area and having limits for the possible ranges they can measure, it is not expected that all

sensors generate valid measurements at all points in time. The array which measures the direction has the longest range, while the range-measuring devices have a shorter reach. This means that the simulated scenario needs to be partitioned into when the different sensors are close enough to measure the UAV.

The division of the measurements is done by inspecting all the UAV position samples and calculating the distance between each sensor and the UAV. If the UAV is within a sensor's feasible radius, the measurements are added to the sensor's measurement list. All samples that are outside are classified as "nan".

3.2 Warehouse Scenario for Navigation of Robots Indoor

The warehouse scenario is made to replicate the AutoStore concept. AutoStore is a high-density, robotic storage and retrieval system designed for warehouse and distribution center operations. It is designed to maximize storage density by stacking packets on top of each other in a grid. The robots are moving on top of that grid guided by a control system that optimizes the storage and retrieval of goods based on real-time data on inventory levels, order volume, and other factors. The robots are tracked using a combination of sensors and cameras that are installed throughout the warehouse, and the information is sent to the control system [48].

3.2.1 Scenario Setup

Since these experiments have not been conducted in a real warehouse environment using actual sensors, it is necessary to create a scenario that closely resembles the circumstances of a real warehouse. This makes the simulation accurately reflect how the navigation system would perform in a warehouse setting.

The warehouse is set to be a room, where the 2D grid the robots are moving on is expected to be of $n \times m$ m^2. The grid covers the whole room, meaning the robots can move over the entire area. Beneath the robots are the stacks of packages, but they are not necessary to consider here since the sensors only need to be able to cover the plane of motion. Independent systems are expected to be used for controlling the packages. With the 2D grid for the robots being set, it is possible to consider the areas where the sensors can be placed. Since the robots move on the ground area, the sensors are limited to at least being placed one meter above them. The maximum height the sensors can have is limited by the roof of the warehouse. If the sensors are placed by the wall or to the roof, they are simply assumed to be attached to the material there. If some sensors are placed in the middle of the room, and not high enough to reach the roof, it is assumed that the sensors are hanging in rods. The available 3D space for the sensors is therefore of size $n \times m \times h$ m^3, where $h = room\_height - 1$ m.

To make the scenario as realistic as possible, it is assumed that some poles are placed in the room to support the roof. The robots are therefore not able to move in the areas where the poles are placed. The radio signals are not able to

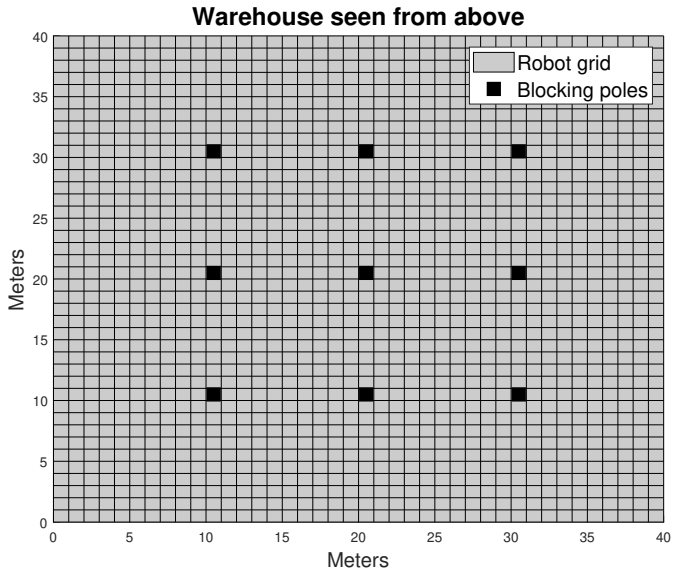


Figure 3.7: A representation of the warehouse as seen from above. Assumed that the robots can move on all the gray squares in the grid.

penetrate through the poles which will limit the reachable distance for sensors close to the poles, hence resulting in an increasing number of sensors needed to cover the entire area.

3.2.2 Scenario Results

The grid on which the robots move on was set to have the dimensions 40×40 m^2. This resulted in 1600 positions on the plane where the robots could move. The space where the sensors could be placed had the same dimensions for width and length as the grid but with an additional three meters in height. The possible height for the sensors was set to be 1 meter to 4 meters above them, where the roof was assumed to be 4 meters above the plane. The sensors needed to be placed at least one meter above the robots to avoid them from being in the robots' paths. This resulted in the 3D grid of $40 \times 40 \times 3$ possible positions of the sensors, which enables a total of 4800 places for the sensors. To mimic a real warehouse, it was assumed that poles were placed in the room to support the weight of the roof. There were a total of 9 poles placed as shown in Figure 3.7, which is an illustration of the warehouse seen from above. Each pole was made with an area of $1 \times 1m^2$ and reaches from the grid to the roof.

Chapter 4

Estimator Design

In this chapter, the methods for evaluating the robustness of using BLE sensors for navigation in the two scenarios created in Chapter 3 will be presented. In order to achieve this, models explaining the different systems must be constructed. To account for the presence of noises and fuse the different measurements together for higher accuracy, an EKF will be used to estimate the desired states in the UAV scenario. The state estimation in the warehouse scenario will be performed through a maximum likelihood estimator. Lastly, the objective is to evaluate the robustness of Bluetooth-based navigation in both cases. This will be accomplished by using the CRLB to calculate the minimum variance of the estimators. Both cases have some accuracy requirements for the estimates which is used when evaluating if the navigation is robust enough in the two cases.

4.1 Estimator Design and Optimization of Net-Landing Scenario

This scenario was created in Section 3.1 and is assumed to take place over water, minimizing objects which may block or reflect the Bluetooth signals. The UAV is assumed to be equipped with an INS. The INS has an IMU that can measure acceleration. The data from the IMU is expected to contain some noises and biases. For additional navigation resources, an antenna array is placed right behind the landing-net. This array is able to measure the direction of the incoming Bluetooth signal and obtain azimuth and elevation values using AoA principles. There will also be a possibility to include a range-measuring sensor together with the antenna array. These measurements are assumed to be independent of each other, as there are two distinct methods for measuring range and directions.

Navigation using only an IMU and one antenna array measurement may not be able to provide satisfactory robustness as it will be vulnerable to only rely on one external sensor. Additional range sensors elsewhere in the simulated area will also be included in the analyses to hopefully improve the robustness. As discussed in Section 2.1.4, they can measure ranges up to a limited distance. Due to their

limited feasible range, the beacons will not be able to measure the UAV's whole trajectory, hence the placement of the beacons decides when the UAV is measured and may affect where the system is robust.

4.1.1 System Model

Since the simulation is made with respect to one frame, the rotation of the UAV is the same for all sensors. This results in the option of using the same frame for all the incoming values. Although this simplification neglects the importance of rotations in a real-world scenario, the inclusion of rotation is not expected to have that big of an effect on the system. The position error is caused by errors in the velocity, as the position is the time derivative of velocity. The velocity is dependent on the orientation, which makes the uncertainty of the orientation affect the velocity which as a result affects the position. The position is still only dependent on the velocity error when neglecting the rotation error [49]. Including the rotation may increase the system's covariance a bit as it is one extra error component that potentially could affect the velocity error and, consequently, the position error. However, neglecting the orientation will not affect the sensor placements effect on the positions covariance. The considered frame is seen from the antenna array, i.e., right by the landing net, and is expressed in NED.

Kinematics

The state of the fixed-wing UAV is explained by

$$\mathbf{x} = [p_N, p_E, p_D, v_N, v_E, v_D, b_{acc,N}, b_{acc,E}, b_{acc,D}]^\top, \tag{4.1}$$

which can be simplified to $\mathbf{x} = [\mathbf{p}, \mathbf{v}, \mathbf{b}_{acc}]^\top \in \mathbb{R}^{9\times 1}$. $\mathbf{p}$ is the position, $\mathbf{v}$ is the linear velocity, and $\mathbf{b}_{acc}$ is the acceleration bias in the IMU, all expressed in the NED frame.

Including bias in the state makes it possible to estimate the accelerometer biases as they may be varying. It also works as an estimator for other forces and errors which are not taken into account in the model but will affect the UAV when flying. The kinematic model can be expressed as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{b}}_{acc} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f} + \mathbf{g} \\ -T_{acc}^{-1}\mathbf{b}_{acc} + \mathbf{w}_{b_{acc}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f}_{acc} - \mathbf{b}_{acc} + \mathbf{g} \\ -T_{acc}^{-1}\mathbf{b}_{acc} + \mathbf{w}_{b_{acc}} \end{bmatrix}, \tag{4.2}$$

which is similar to the kinematic equations in an INS from Section 2.2. $\mathbf{f}$ is the forces acting on the UAV, and is modeled as $\mathbf{f} = \mathbf{f}_{acc} - \mathbf{b}_{acc}$. $\mathbf{f}_{acc}$ is the measured accelerations from the IMU. As mentioned, they are affected by noise and bias, which can be expressed by $\mathbf{f}_{acc} = \mathbf{acc} + \mathbf{b}_{acc} + \mathbf{w}_{\mathbf{acc}}$, where $\mathbf{acc}$ is the true accelerations and $\mathbf{w}_{\mathbf{acc}}$ is additive noise in the measurements. The gravitational constant $\mathbf{g} = [0, 0, 9.81]^\top \mathrm{m/s}^2$ is added to cancel the gravitation from the accelerometer to correctly derive the velocity. T_{acc} is the time-constant referring to how long it

is expected to take before the accelerometer biases drift and is set to 3600s [2]. $\mathbf{w}_{b_{acc}}$ is the assumed errors in the accelerometer's biases.

Writing (4.2) with respect to the states $\mathbf{x}$ from (4.1), classifying the IMU measurements as control input $\mathbf{u} = \mathbf{f}_{acc} + \mathbf{g}$, and the errors as $\mathbf{w} = [\mathbf{w}_{acc}, \mathbf{w}_{b_{acc}}]^\top$ results in

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w}, \tag{4.3}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{T_{acc}}\mathbf{I_3} \end{bmatrix}, \tag{4.4}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0_{3x3}} \\ \mathbf{I_3} \\ \mathbf{0_{3x3}} \end{bmatrix}, \tag{4.5}$$

and

$$\mathbf{E} = \begin{bmatrix} \mathbf{0_{3x3}} \\ \mathbf{I_3} \\ \mathbf{I_3} \end{bmatrix}. \tag{4.6}$$

Writing the equations on this form shows that the kinematic model is linear in $\mathbf{x}$. From the motion equations, it shows that it is possible to determine the UAV's position solely based on the measured acceleration, given that the initial position is known. However, since there are some error factors that affect the resulting states, extra measurements are necessary to avoid a drift over time.

Measurements

There are four sensors that are used to aid the IMU's acceleration measurements to determine the UAV's state. The antenna array placed in local origin generates a bearing vector, $\boldsymbol{\eta}_a$, and a BLE range beacon by the array measures the relative distance from the array, ρ_a. There are two beacons placed somewhere in the grid, p_{b1} and p_{b2}, that measure the relative distances between their position and the UAV, ρ_{b1} and ρ_{b2}, respectively. The bearing measurements are expressed as a unit vector in NED, $\mathbb{R}^{3\times1}$, while the distances are single metrics in meters, $\mathbb{R}^{1\times1}$. The incoming measurements in total can then be expressed as

$$\mathbf{y} = [\eta_{a,N}, \eta_{a,E}, \eta_{a,D}, \rho_a, \rho_{b1}, \rho_{b2}]^\top \in \mathbb{R}^{6\times1}. \tag{4.7}$$

Antenna array

The antenna array measures the direction of the Bluetooth signal. In the same coordinates as the array, there is assumed to be a range-measuring BLE beacon. The direction is given by azimuth and elevation values, ψ_a and α_a respectively.

These values can be combined to express the bearing $\boldsymbol{\eta}_a(\psi_a, \alpha_a)$. As the array is aligned with the NED frame, the azimuth is representing the east angle, while the elevation represents the down angle. The conversion from direction to bearing values is shown in (4.8).

$$\boldsymbol{\eta}_a = \begin{bmatrix} \eta_{a,N} \\ \eta_{a,E} \\ \eta_{a,D} \end{bmatrix} = \begin{bmatrix} \cos(\psi_a) \cdot \cos(\alpha_a) \\ \sin(\psi_a) \cdot \cos(\alpha_a) \\ -\sin(\alpha_a) \end{bmatrix} \tag{4.8}$$

The measurements obtained by both the antenna array and the range beacon are assumed to contain noises. The antenna array measurements can be expressed as

$$y_{\psi_a} = \psi_a + \varepsilon_\psi \tag{4.9}$$

$$y_{\alpha_a} = \alpha_a + \varepsilon_a, \tag{4.10}$$

and the distance measurements can be expressed as

$$y_{\rho_a} = \rho_{a,true} + \varepsilon_{\rho_a}, \tag{4.11}$$

where $\varepsilon_\bullet$ is assumed to be white Gaussian noise. White Gaussian noise is expressed as shown in (4.12), where the PDF has zero mean and $\sigma_\bullet^2$ variance.

$$\varepsilon_\bullet \sim \mathcal{N}(0, \sigma_\bullet^2) \tag{4.12}$$

The additive noises in the direction measurements, (4.9) and (4.10), needs to be taken into account when calculating the bearing, (4.8). The result of this is illustrated in (4.13), where the parameters b_ψ and b_α are used to debias the measurements. This is performed due to the nonlinear mapping from the spherical coordinates of azimuth and elevation to the Cartesian NED coordinates in the model [2]. This is needed since the effect of the errors change when transformed which is necessary to include to obtain reasonable values of the bearing. The parameters are calculated based on the noises in the azimuth and elevation measurements as shown in (4.14) and (4.15).

$$\mathbf{y}_{\eta_a}(\psi_a, \alpha_a) = \begin{bmatrix} b_\psi^{-1} \cdot b_\alpha^{-1} \cdot \cos(\psi_a) \cdot \cos(\alpha_a) \\ b_\psi^{-1} \cdot b_\alpha^{-1} \cdot \sin(\psi_a) \cdot \cos(\alpha_a) \\ b_\alpha^{-1} \cdot \sin(\alpha_a) \end{bmatrix} \tag{4.13}$$

$$b_\psi^{-1} = \mathbb{E}[\cos(\omega_\psi)] = e^{-\frac{\sigma_\psi^2}{2}} \tag{4.14}$$

$$b_\alpha^{-1} = \mathbb{E}[\cos(\omega_\alpha)] = e^{-\frac{\sigma_\alpha^2}{2}} \tag{4.15}$$

BLE beacons

The range measurements taken by the BLE beacons are of the same type as the range measurement taken by the antenna array, as shown in (4.16) and (4.17).

$$y_{\rho_{b1}} = \rho_{b1,true} + \varepsilon_{\rho_{b1}} \tag{4.16}$$

$$y_{\rho_{b2}} = \rho_{b2,true} + \varepsilon_{\rho_{b2}} \tag{4.17}$$

4.1.2 State Estimation with Extended Kalman filter

From the measurement equations, it shows that all the measurements are affected by some noise. Calculating the UAV's state solely based on these measurements will therefore result in erroneous state values. Hence, to get more reliable knowledge about the UAV's position, the incoming measurements are processed via an estimator. The applied estimator is the EKF, which works with the nonlinear nature of the measurement model with respect to the states. The goal of using the EKF is to filter the noises in the measurements, fuse the measurements together, and get results that are robust enough for the navigation case. The choice of implementing the EKF for this scenario is also due to the fact that it works even during loss of measurements. This is valuable as there is no guarantee that the system always receives measurements, as discussed in Section 2.4.

In order to apply the EKF to the presented system, the system equations need to be discretized. This is a result of the measurements being sampled at discrete intervals, hence incoming data are of discrete-time format. The discrete-time quantities are found by using forward Euler integration [50]. The state estimate of $\mathbf{x}$ is then

$$\check{\mathbf{x}}_k \approx \hat{\mathbf{x}}_{k-1} + dT \cdot (\mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}), \tag{4.18}$$

where k is used to express the system at time step k and $k-1$ is the state of the previous interval. dT is the update rate with respect to the control inputs, i.e., the accelerometer's update rate. In (4.18), $\hat{\mathbf{x}}_{k-1}$ is the previous estimated state. $\check{\mathbf{x}}_k$ is the new calculated state solely based on previous estimates and incoming control signals.

The state transition matrix $\mathbf{F}$ is discretized by integration from 0 to dT, resulting in (4.19). The matrix $\mathbf{F}$ is the same as in (4.4) which is time-invariant, hence causing $\mathbf{F}_k$ to be continuous.

$$\mathbf{F}_k = \mathbf{I}_9 + \int_0^{dT} \mathbf{F}\tau \; d\tau \tag{4.19}$$

The noise $\mathbf{w}$ from (4.3) is assumed to be white Gaussian noise with zero mean and $\mathbf{Q}$ covariance matrix. $\mathbf{Q}$ is dependent on the variances of the additive process noises, as shown in (4.20). This matrix then needs to be mapped from the input space to the state space through a similarity transform as shown in (4.21). In this

case, since both the process noise variances and $\mathbf{F}$ are time-invariant, the resulting $\mathbf{Q_k}$ is constant and can be calculated beforehand.

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\sigma}_{acc}^2 & 0 \\ 0 & \boldsymbol{\sigma}_{b_{acc}}^2 \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{4.20}$$

$$\mathbf{Q}_k = \mathbf{F}_k \mathbf{E} \mathbf{Q} \mathbf{F}_k^\top \mathbf{E}^\top \in \mathbb{R}^{9\times 9} \tag{4.21}$$

The measurements can be expressed in two different ways. One is as expressed in equations (4.13), (4.11), (4.16) and (4.17), which is directly related to what the sensors measure. The other method is to calculate the expected measurements by using the estimated position of the UAV and the known position of the sensors. These estimates will be labeled $\hat{y}_\bullet$. For the different range sensors, this can be calculated as shown in (4.22). $\check{\mathbf{p}}_{UAV,k}$ is the position extracted from the estimated state calculated as shown in (4.18).

$$\hat{y}_{\rho_\bullet,k} = \|\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_\bullet\|_2, \tag{4.22}$$

The expected bearing can also be calculated based on the estimated UAV position and the sensor position as

$$\hat{\mathbf{y}}_{\eta_a,k} = \frac{\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_a}{\|\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_a\|_2} \in \mathbb{R}^{3\times 1}. \tag{4.23}$$

The equations for the expected measurements are all nonlinear with respect to the position of the UAV, which, as shown in (4.1), is a part of the state $\mathbf{x}$. To utilize the EKF, it is necessary to linearize the measurement models. The measurement matrices are found by taking the Jacobian of the measurement models with respect to $\mathbf{x}$ about the current state estimate mean. The measurement matrix for the ranges with respect to the position estimates is calculated as shown in (4.24). Note that the partial derivative of $\mathbf{y}$ with respect to the rest of the states, $\mathbf{v}$ and $\mathbf{b}_{acc}$, will be zeros since the measurement models are only dependent on the position states. The total Jacobian matrix for a range measurement is shown in (4.25).

$$\left.\frac{\partial \boldsymbol{y}_{\rho,k}}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k=\check{\mathbf{p}}_{UAV,k}} = \frac{\partial \boldsymbol{y}_{\rho,k}}{\partial \check{\mathbf{p}}_{UAV,k}} = \frac{\left(\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_\bullet\right)^\top}{\left\|\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_\bullet\right\|_2} \tag{4.24}$$

$$\mathbf{G}_{\rho,k} = \begin{bmatrix} \frac{\left(\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_\bullet\right)^\top}{\left\|\check{\mathbf{p}}_{UAV,k} - \mathbf{p}_\bullet\right\|_2} & \mathbf{0}_{1\times 6} \end{bmatrix} \in \mathbb{R}^{1\times 9} \tag{4.25}$$

The measurement matrix for the bearing is calculated as shown in (4.26), which leads to the full $\mathbb{R}^{3\times 9}$ Jacobian matrix shown in (4.27).

$$\left.\frac{\partial \boldsymbol{y}_{\eta,k}}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k=\check{\mathbf{p}}_{UAV,k}} = \frac{\partial \boldsymbol{y}_{\eta,k}}{\partial \check{\mathbf{p}}_{UAV,k}} = \frac{-\mathbf{S}^2(\hat{\mathbf{y}}_{\eta_a,k})}{\left\|\check{\mathbf{p}}_{UAV,k}-\mathbf{p}_a\right\|_2} \tag{4.26}$$

$$\mathbf{G}_{\eta,k} = \left[\frac{-\mathbf{S}^2(\hat{\mathbf{y}}_{\eta_a,k})}{\left\|\check{\mathbf{p}}_{UAV,k}-\mathbf{p}_a\right\|_2} \quad \mathbf{0}_{3\times 6}\right] \in \mathbb{R}^{3\times 9} \tag{4.27}$$

The errors which follow from the range measurements can be expressed in a measurement noise covariance matrix, as shown in (4.28). The measurement noise covariance matrix for the bearing is shown in (4.29).

$$\mathbf{R}_\rho = \begin{bmatrix} \sigma^2_{\rho,a} & 0 & 0 \\ 0 & \sigma^2_{\rho,b_1} & 0 \\ 0 & 0 & \sigma^2_{\rho,b_2} \end{bmatrix} \tag{4.28}$$

$$\mathbf{R}_\eta = \begin{bmatrix} \sigma^2_{\eta,N} & 0 & 0 \\ 0 & \sigma^2_{\eta,E} & 0 \\ 0 & 0 & \sigma^2_{\eta,D} \end{bmatrix} \tag{4.29}$$

An issue which occurs with $\mathbf{R}_\eta$ is that it has dimension $\mathbb{R}^{3\times 3}$, but only depends on the two measurements ψ_a and α_a. The obtained measurement noise covariance matrix from the antenna array is

$$\mathbf{R}_{dir,a} = \begin{bmatrix} \sigma^2_{\psi} & 0 \\ 0 & \sigma^2_{\alpha} \end{bmatrix}, \tag{4.30}$$

which must be mapped into the $\mathbb{R}^{3\times 3}$ bearing noise covariance matrix. The mapping is performed as shown in (4.31). The mapping matrix $\mathbf{M}_\eta$ transforms the measurement noises from spherical to Cartesian coordinates. The calculation of $\mathbf{M}_\eta$ is provided in Appendix A [51].

$$\boldsymbol{R}_\eta = \boldsymbol{M}_\eta \boldsymbol{R}_{dir,a} \boldsymbol{M}_\eta^\top \in \mathbb{R}^{3\times 3} \tag{4.31}$$

With $\mathbf{R}_\eta$ only depending on 2 measurements, the resulting matrix has rank 2, and is therefore only positive semidefinite. This can cause a problem of unrealistic large Kalman gains during the correction step. A method to overcome this issue is to add a scaled matrix to the measurement noise covariance matrix. Equation (4.32) shows the scaling of $\mathbf{R}_\eta$. κ is the scaling value and can be assigned a small value to assure that the matrix is positive definite.

$$\boldsymbol{R}_\eta = \left(\boldsymbol{M}_\eta \boldsymbol{R}_{dir,a} \boldsymbol{M}_\eta^\top + \kappa \cdot \boldsymbol{I}_{3\times 3}\right) > \mathbf{0} \tag{4.32}$$

The total measurement noise covariance matrix can be expressed as shown in (4.33).

$$\mathbf{R} = \text{blkdiag}(\mathbf{R}_\eta, \mathbf{R}_{\rho_a}, \mathbf{R}_{\rho_{b1}}, \mathbf{R}_{\rho_{b2}}) \tag{4.33}$$

With the measurements being independent of each other and not available at all times, the EKF needs to be able to handle missing measurements. This is

done by creating the vectors as shown in (4.34). When some measurements are missing, the corresponding elements in the vectors are left empty.

$$\begin{aligned} \boldsymbol{y} &= \left(\boldsymbol{y}_{\eta}, \boldsymbol{y}_{\rho_a}, \boldsymbol{y}_{\rho_{b1}}, \boldsymbol{y}_{\rho_{b2}}\right)^{\top} \\ \hat{\boldsymbol{y}} &= \left(\hat{\boldsymbol{y}}_{\eta}, \hat{\boldsymbol{y}}_{\rho_a}, \hat{\boldsymbol{y}}_{\rho_{b1}}, \hat{\boldsymbol{y}}_{\rho_{b2}}\right)^{\top} \\ \boldsymbol{R} &= \text{blkdiag}\left(\boldsymbol{R}_{\eta}, \mathbf{R}_{\rho_a}, \mathbf{R}_{\rho_{b1}}, \mathbf{R}_{\rho_{b2}}\right) \\ \boldsymbol{G} &= \left[\boldsymbol{G}_{\eta}, \mathbf{G}_{\rho_a}, \mathbf{G}_{\rho_{b1}}, \mathbf{G}_{\rho_{b2}}\right]^{\top} \end{aligned} \tag{4.34}$$

The above equations can then be used in the EKF algorithm, and by testing and tuning, hopefully yield reasonable results for the estimated states.

4.1.3 Optimization of Sensor Placements

In order to test whether the system is able to provide robust navigation, CRLB is used to give a metric for the lower bound on the variance the EKF estimator is able to achieve given a perfect trajectory. Since the EKF is a Gaussian estimator and the trajectory is known prior to the estimation it is possible to use the parametric CRLB for the Gaussian case as mentioned in Section 2.5 [44]. By this assumption, the CRLB is calculated identically to how the error covariance matrix $\hat{\mathbf{P}}$ is calculated in the EKF, as shown in (4.35). (4.36) and (4.37) represent the Riccati equations to calculate the recursive CRLB, which are the same as used in the EKF.

$$\text{cov}(\hat{\mathbf{x}}) \succeq \mathcal{I}^{-1}(\mathbf{x}) = P_{\mathbf{x}}^{\text{CRLB}} \tag{4.35}$$

$$\mathbf{P}_{k+1|k}^{CRLB} = \mathbf{F}_k \mathbf{P}_{k|k}^{CRLB} \mathbf{F}_k^{\top} + \mathbf{Q}_k \tag{4.36}$$

$$\mathbf{P}_{k+1|k+1}^{CRLB} = (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k) \mathbf{P}_{k+1|k}^{CRLB} (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k)^{\top} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\top} \tag{4.37}$$

In order to utilize the EKF to calculate the CRLB instead of estimating the states, the true states from the simulated trajectory are used as state inputs instead of the previously estimated states. Therefore, all the expressions in the algorithm are evaluated based on the true trajectory. This will result in a different $\mathbf{G}_k$, as it in the EKF was linearized around $\hat{\mathbf{x}}_k$, but will here be linearized around the true $\mathbf{x}_k$. The sensor measurements still contain the expected noises and errors, causing the resulting covariance matrix to be solely based on how the filter is able to perform given true states with faulty measurements. This equals a case where the estimator is perfect, i.e., representing the best results it is possible to obtain with the EKF for this system.

Having a method for calculating the CRLB, it is possible to inspect how precisely the filter can behave for the perfect case. The varying factor which affects the resulting CRLB is the beacon positions. Since their positions decide when the UAV is measured through its trajectory, they affect the accuracy of the estimator. To achieve positions that give acceptable accuracy for the scenario, an optimization algorithm is applied.

Algorithm 2 CRLB minimization algorithm, simplified

Initialize:
$n_{min} \geq N \geq n_{max}$, $e_{min} \geq E \geq e_{max}$ and $d_{min} \geq D \geq d_{max}$
Create $positions$ which is a meshgrid of possible positions
$P_{min} \leftarrow inf$ ▷ Set initial CRLB matrix to inf
$best\_pos \leftarrow [0\ 0\ 0; 0\ 0\ 0]$ ▷ Matrix to save the best beacon positions
for $i, j = 1$ to number of possible positions **do**
  $p_{b1} = positions(i,:), p_{b2} = positions(j,:)$
  Calculate distance d between p_{b1} and p_{b2}
  if $d \geq 10m$ **then**
    Separate measurements with respect to the new positions
    Run EKF algorithm to calculate CRLB, returns P^{CRLB}, the trace of the position's state covariance
    if $P^{CRLB} < P_{min}$ **then**
      $P_{min} \leftarrow P^{CRLB}$
      $best\_pos \leftarrow [p_{b1}; p_{b2}]$
    end if
  end if
end for

The algorithm iterates through a grid of all possible positions. The positions have upper and lower bounds based on where in the frame they can be placed. To reduce the run time of the algorithm, a step size between each possible position is set, and the algorithm only runs given that the two positions are a certain distance away from each other. For all the possible combinations, the measurements from the simulations need to be separated based on when the sensors are in the range of the UAV, then the EKF algorithm which calculates the CRLB is executed. The chosen CRLB evaluation value is then the same as the diagonal of the error covariance matrix $\mathbf{P}^{CRLB}$. In the EKF algorithm, there are tests that analyze the resulting covariance matrix. The tests are only looking at the trace of the elements corresponding to the UAV's position since this is a factor of importance. The trace is returned as the CRLB of the position estimation throughout the trajectory. A pseudo-code of the minimization algorithm implemented is shown in Algorithm 2. After iterating through all the positions, it will return the positions that yielded the lowest trace of the covariance matrix.

4.1.4 Performed Experiments

When experimenting with the scenario it is interesting to see how the accuracy is changing throughout the trajectory based on what the optimization goal is and how the BLE beacons can be used to help achieve the goal.

First, a scenario with only the antenna array will be tested. Optimization is not possible in this case, since the antenna array is assumed to have a fixed posi-

tion. The estimation will then be performed using only the bearing measurements from the array and the INS. Having this as an initial estimation, it is possible to compare the results of later estimations where additional beacons are used and their placements have been optimized.

In order to test how additional beacons affect the navigation accuracy, one beacon is assumed to be placed by the array, and two beacons are available for positioning elsewhere. The trajectory of the UAV is planned to start far away from the landing-net which means that the beacons are unable to measure the range at all times. The optimization is therefore focused on how the two beacons should be placed in order to fulfill different requirements. There are different ways the optimization may be handled based on what the wanted outcome is. One optimization technique will focus on finding the beacon positions which give the overall lowest CRLB throughout the trajectory. However, the positions which give the best accuracy overall do not have any guarantee that the accuracy when the UAV is close to the landing-net will be good. For this reason, other optimizations will be also performed, where some constraints assuring the accuracy close to the landing-net have been included.

There will be two different techniques for optimizing the positions while assuring good accuracy close to the landing-net. The first technique will solely focus on the accuracy of the estimation when the UAV is moving closer to the landing net. This means that the algorithm will only compare the CRLB value when it is closer than a given distance, and choose the positions with the best accuracy close. The accuracy when the UAV is close will then be the best it can be, but there is no control over how good the estimation will be for the trajectory far away.

The second technique will try to focus on both good accuracies close to the net while still maintaining an acceptable accuracy far away. This is done by finding the trajectory with the lowest CRLB peak, but only taking it into account if the CRLB is below a given accuracy when the UAV is close.

The three distinct methods of optimizing the problem can result in varying outcomes for both the accuracies and the positions of the beacons. The most optimal solution may differ based on the specific use case.

4.2 Estimator Design and Optimization of Warehouse Scenario

When considering scenarios like AutoStore, a potential approach for locating the robots involves utilizing Bluetooth sensors. By utilizing a combination of range sensors and direction finding sensors, it is possible to explore the level of precision in the positioning that can be achieved. However, there is a trade-off between the desired level of accuracy and the number of sensors that can be employed to minimize costs. The optimization problem revolves around finding the optimal placements with the lowest number of sensors required to achieve sufficient accuracy.

This section of the thesis focuses on exploring whether the accuracy of navigating the robots in a warehouse scenario can be adequate using BLE sensors. The accuracy will be measured with respect to how well the sensors can localize each point in the grid, assuming that sufficient localization accuracy provides an adequate foundation for a navigation system. Given the existence of similar warehouses in reality, it is worth exploring whether implementing BLE sensors can potentially reduce the cost of the navigation system compared to the current system.

4.2.1 Sensor Measurements Model

The Bluetooth devices can either be direction finding devices or BLE beacons for range estimation. The sensors measure the ranges and angles the same way as in the net-landing case, Section 4.1. The measured range contains an error ε_ρ as shown in (4.38), and the angles ψ and α are combined to express the bearing as shown in (4.39). As in the previous scenario, the bearing has to be debiased due to the noises present in the azimuth and elevation, as shown in (4.40)

$$y_\rho = \rho_{true} + \varepsilon_\rho \tag{4.38}$$

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_N \\ \eta_E \\ \eta_D \end{bmatrix} = \begin{bmatrix} \cos(\psi)\cdot\cos(\alpha) \\ \sin(\psi)\cdot\cos(\alpha) \\ -\sin(\alpha) \end{bmatrix} \tag{4.39}$$

$$\mathbf{y}_\eta(\psi,\alpha) = \begin{bmatrix} b_\psi^{-1}\cdot b_\alpha^{-1}\cdot\cos(\psi_a)\cdot\cos(\alpha_a) \\ b_\psi^{-1}\cdot b_\alpha^{-1}\cdot\sin(\psi_a)\cdot\cos(\alpha_a) \\ b_\alpha^{-1}\cdot\sin(\alpha_a) \end{bmatrix} + \boldsymbol{\epsilon}_\eta \tag{4.40}$$

The precision of the sensors used in the warehouse directly affects the accuracy of the robots' localization. The noises in both the distance and the direction-finding solution are assumed to be white Gaussian noise. The noises can be expressed on the form

$$\varepsilon_\bullet \sim \mathcal{N}(0,\sigma_\bullet^2). \tag{4.41}$$

Given multiple measurements for one position, the variances can be combined into a noise covariance matrix as shown in (4.42), where u is the number of available range measurements and v is the number of available direction measurements. The range measurements is one dimensional $\boldsymbol{\epsilon}_\rho \in \mathbb{R}^{1\times 1}$ while the bearing is of three dimensions $\boldsymbol{\epsilon}_\eta \in \mathbb{R}^{3\times 3}$. The mapping performed in the net-landing scenario to transform from spherical to Cartesian coordinates to obtain the bearing covariances is not possible in this case. The mapping matrix was dependent on the angle measurements which are not obtained in this scenario. However, according to [52, p. 255] there are cases where it is possible to assume the approximation $\mathbf{R}_\eta \approx \mathbf{I}_3\sigma_\eta^2$, creating a nonsingular matrix. σ_η^2 is the assumed variance from the direction finding sensors. This assumption on the bearing noises is used, hence avoiding the need for mapping and ensuring that $\mathbf{R}_\eta > \mathbf{0}$.

$$\mathbf{R} = \text{diag}(\begin{bmatrix} \sigma^2_{\rho,1} & \dots & \sigma^2_{\rho,u} & \boldsymbol{\sigma}^2_{\eta_1} & \dots & \boldsymbol{\sigma}^2_{\eta_v} \end{bmatrix}) \tag{4.42}$$

The measurements in each position can be expressed as

$$\mathbf{y}_i = \mathbf{g}(\boldsymbol{\theta}_i) + \mathbf{w}_i, \tag{4.43}$$

where $\boldsymbol{\theta}_i = (x_i, y_i, z_i)$ is position number i in the grid, and $\mathbf{g}$ contains the measurements function corresponding to the available sensors. z_i in $\boldsymbol{\theta}_i$ will always remain zero since the grid is placed aligned with the z-axis, but is included due to the 3D positions of the sensors. $\mathbf{w}_i \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_i)$ is the additive Gaussian noises with respect to the measuring sensors. The functions in $\mathbf{g}$ can take two forms, depending on what the corresponding sensor measures. Assuming available range measurements, the function has the form

$$g_\rho = ||\boldsymbol{\theta} - \mathbf{p}_\rho||_2 \in \mathbb{R}^{1\times 1}, \tag{4.44}$$

which is the distance between the range sensor and the position in the grid. The bearing measurements are expressed as

$$\mathbf{g}_\eta = \frac{\boldsymbol{\theta} - \mathbf{p}_\eta}{\|\boldsymbol{\theta} - \mathbf{p}_\eta\|_2} \in \mathbb{R}^{3\times 1}, \tag{4.45}$$

resulting in a 3-dimensional vector that explains the angles of where the robot is with respect to the sensors in the warehouse frame.

The total measurement function $\mathbf{g}$ is varying depending on the sensors, and can be written as

$$\mathbf{g} = \begin{bmatrix} g_{\rho,1} \\ \vdots \\ g_{\rho,u} \\ \mathbf{g}_{\eta,1} \\ \vdots \\ \mathbf{g}_{\eta,v} \end{bmatrix} \in \mathbb{R}^{(u+3v\times 1)}, \tag{4.46}$$

with u numbers of ranging sensors and v numbers of direction sensors. Performing an iteration to go through all the possible positions in the grid, the measurement function will change depending on the sensors which are able to reach the different grid positions.

4.2.2 Calculation of Localization Accuracy in every Grid Point

In this scenario, it is only needed to check one position at a time, not the coverage over a trajectory. This is to ensure that the entire grid will be sufficiently covered, hence crating reliable navigation in the entire warehouse. To achieve this, it is possible to use the CRLB to calculate the best accuracy achievable in each of the points in the grid. This results in the calculation of a single point each time, not

a recursive algorithm that is dependent on the CRLB in the previous iteration. The CRLB is found by using the available information about the likelihood of a position.

The likelihood of obtaining position $\boldsymbol{\theta}_i$ given the available measurements in position i takes the form

$$\mathcal{L}(\boldsymbol{\theta}_i \mid \mathbf{y}_i) = p(\mathbf{y}_i \mid \boldsymbol{\theta}_i) = \mathcal{N}(\mathbf{y}_i; \mathbf{g}(\boldsymbol{\theta}_i), \mathbf{R}_i). \tag{4.47}$$

where $\mathbf{R}_i$ is the measurement noise covariance matrix based on which sensors are available and their variances, as stated in (4.42).

With the likelihood available, it is possible to calculate the FIM as (2.22) states. $\boldsymbol{\theta}$ is from here on used instead of $\boldsymbol{\theta_i}$ for a clearer expression. The FIM, $\mathcal{I}(\boldsymbol{\theta})$, can be derived according to

$$\begin{aligned}
\mathcal{I}(\boldsymbol{\theta}) &= E\left[(\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y} \mid \boldsymbol{\theta}))(\nabla_{\theta} \ln p(\mathbf{y} \mid \boldsymbol{\theta}))^{\top}\right] \\
&= E\left[\mathbf{G}^{\top}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{g}(\boldsymbol{\theta}))(\mathbf{y}-\mathbf{g}(\boldsymbol{\theta}))^{\top}\mathbf{R}^{-1}\mathbf{G}\right] \\
&= \mathbf{G}^{\top}\mathbf{R}^{-1}E\left[(\mathbf{y}-\mathbf{g}(\boldsymbol{\theta}))(\mathbf{y}-\mathbf{g}(\boldsymbol{\theta}))^{\top}\right]\mathbf{R}^{-1}\mathbf{G} \\
&= \mathbf{G}^{\top}\mathbf{R}^{-1}\mathbf{G},
\end{aligned} \tag{4.48}$$

where $\mathbf{G}$ consist of $\mathbf{G}_{\rho}$ and $\mathbf{G}_{\eta}$, which are the Jacobians of the measurement functions $\mathbf{g}_{\bullet}$, given as

$$\mathbf{G}_{\rho} = \frac{\partial \boldsymbol{g}_{\rho}}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{g}_{\rho}}{\partial \boldsymbol{\theta}} = \frac{\left(\hat{\boldsymbol{\theta}} - \mathbf{p}_{\rho}\right)^{\top}}{\left\|\boldsymbol{\theta} - \mathbf{p}_{\rho}\right\|_2} \in \mathbb{R}^{1\times 3} \tag{4.49}$$

$$\mathbf{G}_{\eta} = \frac{\partial \boldsymbol{g}_{\eta}}{\partial \boldsymbol{\theta}_k} = \frac{\partial \boldsymbol{g}_{\eta}}{\partial \boldsymbol{\theta}} = \frac{-\mathbf{S}^2(\mathbf{g}_{\eta})}{\left\|\boldsymbol{\theta} - \mathbf{p}_{\eta}\right\|_2} \in \mathbb{R}^{3\times 3}. \tag{4.50}$$

The resulting simplicity of (4.48) is due to the noises being additive Gaussian, resulting in a special case, mentioned in [43].

The CRLB can then be calculated by taking the inverse of the FIM as in (4.51). To obtain the positioning accuracy for the entire robot grid, an iteration goes through all possible robot positions and calculates the CRLB in each step. For each position, the output value from the CRLB calculation will be stored in a map with the same size as the grid. For each position in the grid, the corresponding value in the map will explain the attainable precision at that point.

$$CRLB_{\boldsymbol{\theta}} \geq \mathcal{I}(\boldsymbol{\theta})^{-1} \tag{4.51}$$

4.2.3 Optimization of sensor placements

The variable factors in this scenario are the number of sensors and their placements in the warehouse. It is desirable with an optimization algorithm that finds a solution with acceptable accuracy over the whole grid while keeping the number of sensors low.

The optimization algorithm created for this cause utilizes a greedy search method to determine the optimal sensor placements. A greedy algorithm is able to arrive at an optimal solution quicker than a dynamic-programming method. The greedy method is more efficient since it accepts the best possible choice at the moment, instead of testing every possible solution before deciding [53]. The first step is to acquire suitable positions for an initial guess of the number of sensors. With an assumption that the sensors can be positioned anywhere over the same area as the $n \times m$ m^2 grid, and $h-1$ meters above the grid of robots, the resulting grid of potential sensor placements is huge. In the previous net-landing scenario, the optimization process only involved moving two sensors, allowing for an exhaustive search of all feasible placements without creating a significantly large optimization problem. However, in this current scenario, a larger number of sensors are required. If every possible combination of sensor placements were to be explored, the optimization problem would be too large, potentially resulting in impractical computational time. To address the issue of problem size, the algorithm adapts a randomized approach where the sensors are positioned randomly in the room and then tested to determine if they provide sufficient coverage of all points.

Sufficient coverage is decided by how many sensors are within reach for a single point in the grid. Since the robots can move over all the possible positions on the grid except through the poles, it is necessary that each possible point is covered by a given number of sensors. Since one sensor is not able to obtain a robot's location information by itself, there need to be at least 3 sensors. The reason for this is to have enough measurements to use trilateration for position estimation at each point. Distance sensors localization may be sufficient with only 2 sensors, but 3 is needed if only range sensors are used, cf. Section 2.1.5. This is solved by creating an $n \times m$ dimensional map, where n and m correspond to the size of the robots' grid. All the values in the map are initialized to 0. For each sensor that is placed in the room, the areas which the sensor is covering are incremented by 1. This continues until all the sensors are placed. If all the points in the map have a value of 3 or more, the sensor placements are sufficient for further testing. Otherwise, new randomized placements need to be explored. The algorithm continues to try new sensor placements until all the areas in the grid are covered.

An additional check which needs to be performed when placing the sensors is to take the poles placed in the room into account. Even if a point is within a sensor's range, it may not be covered by the sensors if there is a pole between the points. A check on whether the pole blocks a signal or not is therefore performed. Given a pole between the sensor and a grid point, the point will not be marked as covered by that sensor.

Given sufficient cover, the algorithm iterates through all the possible robot positions and calculates a CRLB value for each point. The goal of the optimization is to find a viable solution that meets the specified constraints. If such a solution is found, the algorithm stops rather than continuing the search. An extensive search could lead to other sensor placements which might offer a slightly better accur-

acy. However, by having a constraint on the desired level of accuracy, this method ensures that the chosen sensor placements are meeting the constraints within a reasonable search time. It also aims to find the lowest number of sensors needed. This is performed by starting the optimization with only a small amount of available sensors. If the algorithm has performed many iterations in order to find a sufficient cover of the grid without success, the number of sensors increases by 1. The search for sufficient placements then starts over again until the constraints are met. Sufficient positions do not ensure sufficient accuracy, hence if the accuracy is too low, the algorithm needs to start over with finding other sensor positions. The optimization process is described using pseudo-code in Algorithm 3. The algorithm is a simplification, however, it contains the key elements.

Algorithm 3 Optimization of sensors placement in warehouse scenario

Initialize:
Create $[m \times n]$ matrix of robot grid
Create $[m \times n \times h]$ possible sensor position
Initialize number of sensors
$covered \leftarrow 0 \cdot [m \times n]$ ▷ Count number of sensors covering each point
$CRLB\_grid \leftarrow \cdot [m \times n]$ ▷ To save the CRLB for each point
$max\_CRLB \leftarrow \lambda$ ▷ Set λ to wanted CRLB limit
while Not sufficient cover or Not sufficient CRLB **do**
  if Iteration > max_iterations **then**
    Increment number of sensors by 1
  end if
  Randomly place the sensors
  Iterate through all points to see how many sensors they are in the range of
  if Covered < 3 **then**
    Not sufficient cover
  end if
  for All points in robot grid **do**
    Create g and R based on sensors available
    Calculate Jacobian G
    Calculate the CRLB in that point
    Insert calculated CRLB to $CRLB\_grid$ at same position as the point
  end for
  if Any point in $CRLB\_grid < max\_CRLB$ **then**
    Not sufficient CRLB
  end if
  Increment number of iterations by 1
end while

4.2.4 Performed experiments

In order to see how the planned system works and how the different sensors affect the accuracy, multiple cases of this scenario will be experimented with. First, there will be conducted a test where only range sensors are placed in the room to see how many sensors are needed in order to obtain the wanted accuracy of the whole room. Then the same test will be performed using only direction sensors. Then these two will be combined, in order to see if the number of sensors will be reduced given the same limit on the accuracy. From theory, Section 2.1.4, the direction sensors have a higher accuracy when measuring, so an interesting factor will be how the accuracy is changing if only the range or direction is used, and if the case where both are combined gives the best result. Because of the randomization in the optimization algorithm, the solution may differ between runs. Multiple optimizations for each test will therefore be performed, where the results from each run are saved and discussed.

The last test is to adjust the accuracy constraints. Instead of requiring a certain CRLB value that all the points need to be lower than, an option is to set a limit that only a certain percentage of the points in the grid are covered. It may not be necessary that all the points are perfectly covered if many of the points around it have good enough accuracy. If constraining the accuracy in this way results in fewer sensors, it may be a valid method in some use cases where strict accuracy over the whole plane is excessive.

An interesting factor here is whether the accuracy can be kept high while the number of sensors is low. Since the current sensor system is working as it should in a warehouse, it is important to see if the cost can be lowered by utilizing BLE devices instead of the current equipment.

Chapter 5

Results

5.1 Results from the Net-Landing Scenario

5.1.1 Estimation with only Antenna Array Measurements

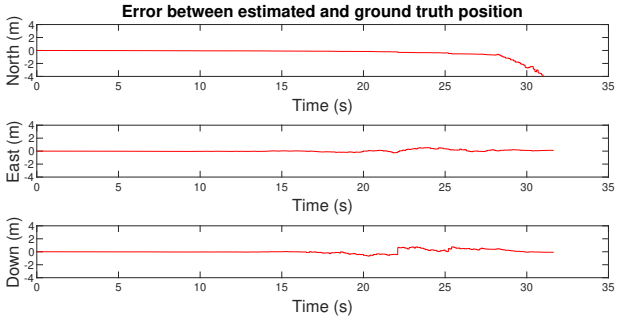
To see how the initial system with only one measuring element would perform, only the direction finding antenna array was used to measure in the first test. An IMU was present on the UAV, providing the process with internal acceleration measurements. The erroneous antenna array directions and acceleration measurements were presented in Section 3.1, in Figure 3.5 and Figure 3.6, respectively. The directions were transformed into bearing measurements as stated in (4.13). The test was performed by estimating the UAV's state with an EKF as expressed in Algorithm 1.

The initial error covariance matrix was initialized to

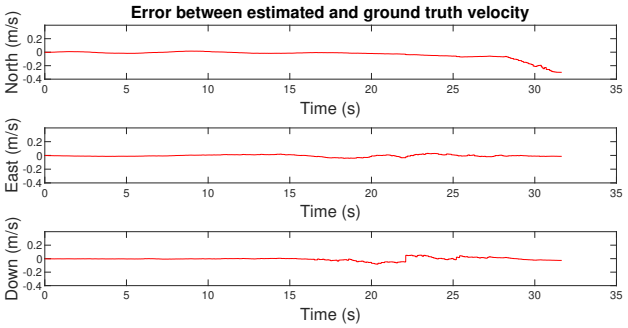
$$\mathbf{P}_0 = \text{diag}([0.1,\ 0.1^2,\ 0.01 \cdot (50 \cdot 9.81/1000)^2]), \tag{5.1}$$

which were obtained by tuning after an initial guess based on known errors. Each of the numbers in the matrix is three-dimensional, corresponding to the NED axis, resulting in a 9×9 matrix.

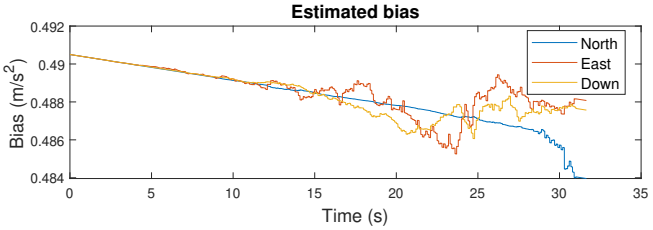
The estimation of the UAV's state was executed and the results after iterating over the whole trajectory are shown in Figure 5.1. The error plots display an increasing error in the north values in both the position and velocity states. The east and down errors contained little visible uncertainty, but some sudden jumps are present in the estimation. However, the east and down position and velocity errors remained mostly around zero. The variance of the position estimation was taken from the resulting error covariance matrix $\hat{\mathbf{P}}$. Extracting the position variances and taking the square root gives the standard deviations, which are shown in Figure 5.1d. The standard deviation of the north position had an increasing trend, while east and down appeared to reduce the uncertainty when the UAV moved closer to the net. The estimated biases appeared to be more varying throughout the trajectory, as displayed in Figure 5.1c.



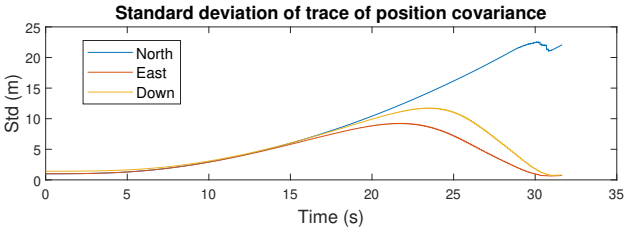
(a) Position estimation error.



(b) Velocity estimation error.



(c) Bias estimates.



(d) Component-wise standard deviation of position estimation using only antenna array measurements.

Figure 5.1: Various plots of the results after estimation using only bearing measurements.

Table 5.1: Resulting beacon positions with corresponding CRLB peak values after optimization. $\sigma^{CRLB} = \sqrt{P^{CRLB}}$

	σ^{CRLB}_{max}	$\sigma^{CRLB}_{max} < 30m$	Beacon Positions	
$\text{CRLB}_{overall}$	8.004m	1.920m	$[80, 50, -50]$	$[200, 10, -40]$
CRLB_{close}	12.215m	0.655m	$[40, -50, -50]$	$[40, 50, -50]$
CRLB_{both}	8.004m	1.920	$[80, 50, -50]$	$[200, 10, -40]$

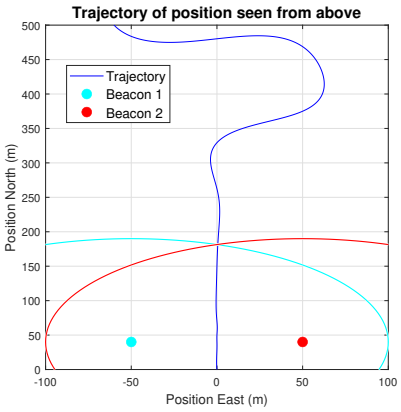
5.1.2 Optimization of Sensor Placements

The optimization algorithm, Algorithm 2, has a run time that is very dependent on the number of possible sensor positions since the EKF has to be executed for each possible combination of positions. To reduce the run time, different actions were performed. The grid in which the UAV is flying was set to be $500 \times 100 \times 50$ meters, as seen in Figure 3.1. However, it was assumed that the beacons were not placed any further than 200 meters north, substantially reducing the number of positions. In addition to this, different step intervals were specified when creating the grid. The north coordinates were 40 meters apart, the east coordinates 20 meters apart, and the vertical positions were separated by 10 meters. This reduced the search space and prevented the search from exploring positions very close to each other. If the beacons were placed nearby to each other, the geometry of the measurements would be bad, and most likely not give good CRLB values. The resulting CRLB values will be expressed by the standard deviations. The reason for this is that explaining the error in meters instead of meters squared gives more understandable values.

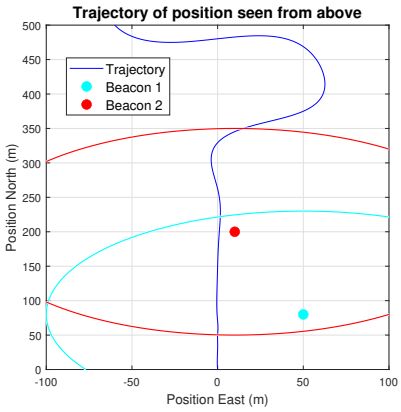
Table 5.1 indicates that both the solution which prioritizes the overall best CRLB value, $CRLB_{overall}$ and the solution aiming for a low CRLB overall as well as close to the net, $CRLB_{both}$, yielded the same beacon positions. The CRLB in these cases had a maximum peak with a standard deviation of 8.004m. However, the accuracy close to the landing net still achieved a quite low standard deviation of 1.920m.

The achieved CRLB peak when solely focusing on obtaining the absolute minimal variance when the UAV was within 30 meters of the landing net, $CRLB_{close}$, had a standard deviation peak of 12.215m. This is a much larger standard deviation than $CRLB_{best}$, which was 8.004m. On the other hand, the accuracy close to the net had a maximum CRLB value of only 0.655m. This sub-meter level of accuracy when the UAV is moving closer to the landing net may be preferable.

The beacon positions from the optimizations are illustrated in Figure 5.2. It is evident from $CRLB_{close}$ that the beacons are positioned symmetrically with respect to the antenna array, as seen in Figure 5.2a. By the illustrated feasible radii, both beacons start to measure the UAV's trajectory simultaneously. Figure 5.2b shows that in $CRLB_{overall}$ and $CRLB_{both}$, beacon 2 is placed further from the antenna array than beacon 1. Measurements are therefore available from beacon 2 earlier



(a) Beacon positions from $CRLB_{close}$ optimization



(b) Beacon positions from $CRLB_{overall}$ and $CRLB_{both}$ optimization

Figure 5.2: Illustrations of the different beacon positions and the expected UAV trajectory. The corresponding circle illustrates the feasible range of the same colored beacon.

than from beacon 1. In addition, both beacons are placed east of the antenna array.

The evolving standard deviations for the three cases are plotted together in Figure 5.3. It can be seen that $CRLB_{overall}$ and $CRLB_{both}$ had the same results as they are overlapping. From here on, only $CRLB_{overall}$ will be discussed for simplicity. The higher peak of $CRLB_{close}$ is clearly demonstrated in the figure. Both cases have an increasing trend in the CRLB value, but $CRLB_{overall}$ seems to improve its estimation earlier than $CRLB_{close}$ as the standard deviation starts to decrease at about 16 seconds. The first jump in $CRLB_{overall}$ reduced the variance slightly, but a more significant drop happened at about 22 seconds. The error in $CRLB_{close}$ continued to increase longer than $CRLB_{overall}$, but when it first dropped, the change was sudden and substantial. Both cases resulted in quite low uncertainties for the last 8 seconds.

To test whether the beacon positions from the optimizations actually resulted in reasonable navigation solutions, the EKF algorithm was executed with the various positions. Both the results from $CRLB_{overall}$ and $CRLB_{close}$ were tested to see how the EKF estimation was affected by the difference in expected variance.

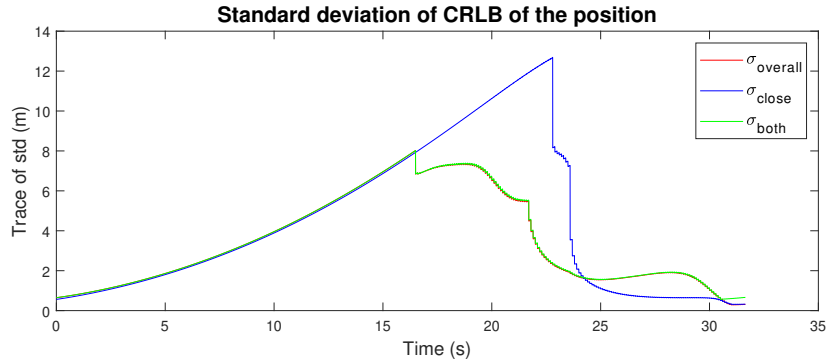


Figure 5.3: The three results of CRLB standard deviation traces plotted together. $\sigma_{overall}$ and σ_{both} are equal.

EKF performed with Results from $CRLB_{overall}$

To execute the EKF with the results from $CRLB_{overall}$, the Bluetooth beacon positions needed to be set with respect to the resulting positions as mentioned in Table 5.1. With the beacon positions in this case, the resulting noisy range measurements are shown in Figure 5.4. The figure shows that the first range measurement came from beacon 2 and started after about 16.5 seconds. Beacon 1 was placed so it started to measure from about 22 seconds, and the UAV was within range of the landing net sensor at about 23.5 seconds. The bearing measurements and incoming acceleration measurements were the same as in the previous test, as shown in Figure 3.5 and Figure 3.6, respectively. By comparing the jumps in CRLB values in Figure 5.3 with when the different measurements started, it can be seen that the first change in variance happened when the first range measurements started. The second jump was caused by the second beacon starting to provide measurements.

The results after executing the EKF algorithm are shown in Figure 5.5. Figure 5.5a shows the estimation error in each of the NED positions. The position estimation appears quite good in the first 16 seconds. After 16 seconds, the plot displays a lot of sudden errors. This was when the first Bluetooth range sensor

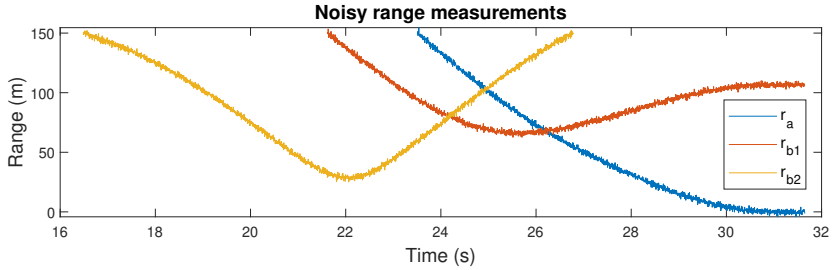
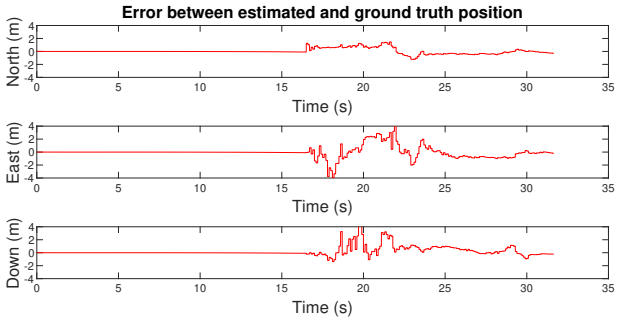


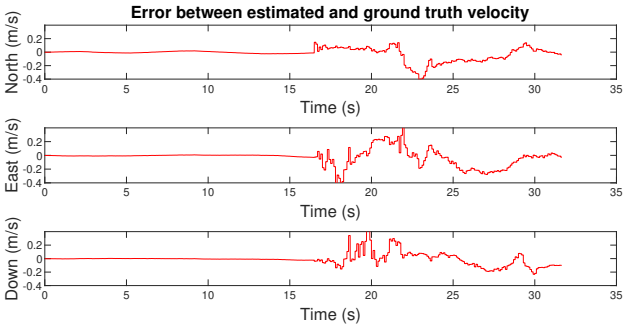
Figure 5.4: Range measurements when placing beacons wrt. the $CRLB_{overall}$ results.

started to provide measurements. The errors are larger compared to when only using bearing, but it appears to iterate around zero. The errors also appear to reduce after approximately 24 seconds, when all three range measurements were available. Some increasing errors can be seen in the position errors after about 27 seconds, which was when beacon 2 got out of range and stopped measuring. The same behavior is noticeable in the velocity error plots, as seen in Figure 5.5b. Figure 5.5c shows the resulting estimated biases, which appear to be a lot more varying than in Figure 5.1c. These changes also appear to start at the same time as the first beacon starts to measure range.

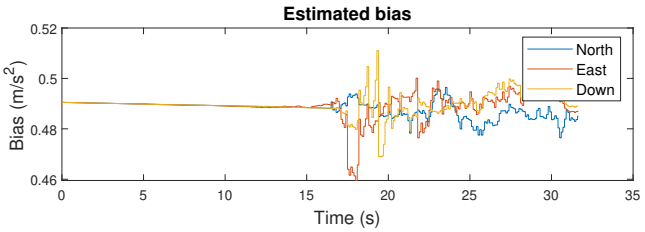
The last plot from the EKF is the actual obtained standard deviation values of the estimation from the error covariance matrix. The actual peak of variance of the estimation resulted in a standard deviation of 12.056m which is much higher than anticipated in the CRLB calculation for the optimal trajectory which was 8.004m.



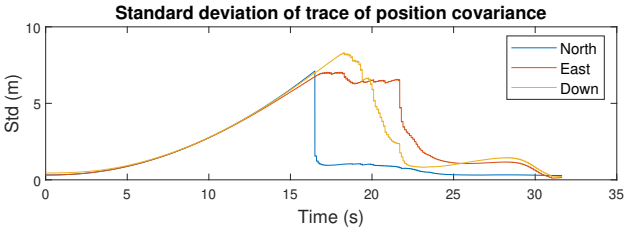
(a) Position estimation error.



(b) Velocity estimation error.



(c) Bias estimates.



(d) Component-wise standard deviation of position estimation.

Figure 5.5: Various plots of the results after EKF performed with using beacon positions from $CRLB_{overall}$.

EKF performed with Results from $CRLB_{close}$

The only factor that changed here concerning the $CRLB_{overall}$ case was the beacon positions. The resulting noisy range measurements are shown in Figure 5.6. It only appears to be two range measurements available, but as the beacons are placed symmetrically to each other, they measured almost the same range at all times. As a consequence, the measurements from beacon 1 and beacon 2 are overlapping in the plot. The plot shows that the planned beacon positions resulted in all the range measurements being available relatively late in the trajectory. The time the sensors started measuring corresponds to the same time the variance from the CRLB estimation was expected to rapidly fall, from Figure 5.3.

The EKF was executed with the same initial error covariance matrix as before, (5.1). The results of the estimation are shown in Figure 5.7. The estimation error in the NED positions can be seen in Figure 5.7a. In all directions, there are some rapid-changing errors starting at approximately 23 seconds. This was the time when the two beacons started measuring. At about 24 seconds, the last range sensor was available, and the errors got instantly smaller. The error remained stable and small for the rest of the estimation. The velocities, Figure 5.7b, also displayed some increasing errors when the ranges were available. The estimated biases acted the same way as in the $CRLB_{overall}$ case, by starting to show large variations when the range measurements were available, shown in Figure 5.7c. The actual standard deviations from the error covariance matrix of the estimation are shown in the last plot Figure 5.7d. The CRLB calculated standard deviation peak for this case was 12.215m, while the actual obtained standard deviation peak of the completed estimation was 19.971m. The estimation therefore ended up with almost twice the standard deviation, however, it can be seen from the plot that the estimation close to the net still was able to obtain low standard deviations. Looking at the standard deviation from the EKF of $CRLB_{overall}$ in comparison with this, this case was still able to obtain a good accuracy faster, and remained good for the rest of the estimation.

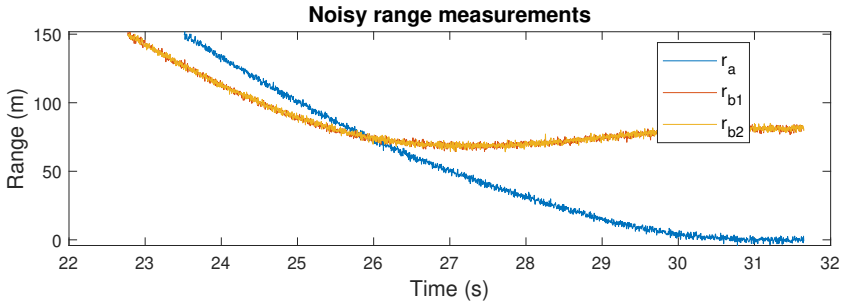
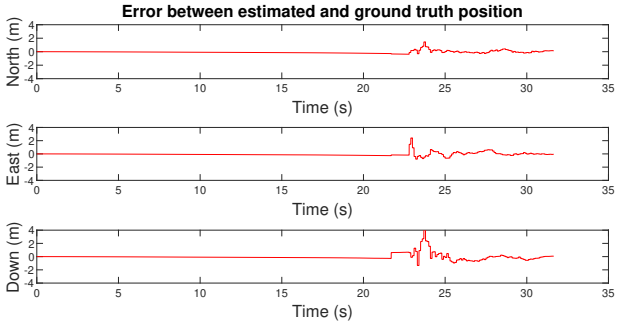
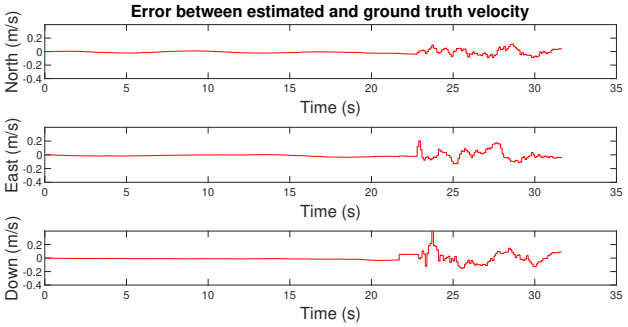


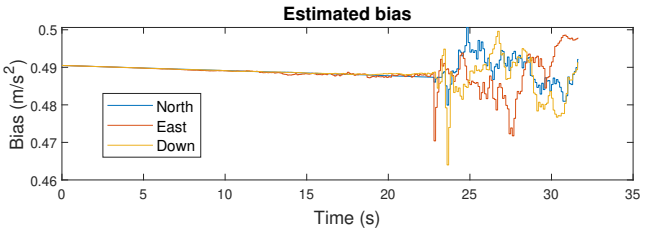
Figure 5.6: Range measurements when placing beacons wrt. the $CRLB_{close}$ results.



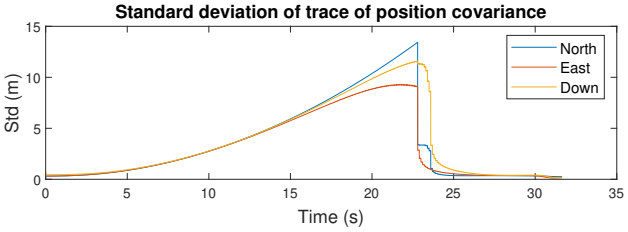
(a) Position estimation error.



(b) Velocity estimation error.



(c) Bias estimates.



(d) Component-wise standard deviation of position estimation.

Figure 5.7: Various plots of the results after EKF performed with using beacon positions from $CRLB_{close}$.

5.2 Results from Warehouse Scenario

The only variables which needed to be set to calculate the CRLB of the position estimation were the different expected variances and the maximum range the sensors could measure. The expected variances in the range measurements were set to

$$\sigma_\rho^2 \approx 2\text{m}^2, \tag{5.2}$$

and the bearing variance was taken from [18], and expected to be

$$\sigma_\eta^2 \approx 1.2^\circ. \tag{5.3}$$

The corresponding limits on the distances were set to be

$$d_\rho = 20\text{m and } d_\eta = 50\text{m}, \tag{5.4}$$

which are less than the maximum feasible distances with respect to the mentioned values in theory, Section 2.1.4. However, they were reduced due to the indoor limitations. If the distances were set to measure up to 100m, the radio signals would most likely be reflected or blocked by something during propagation.

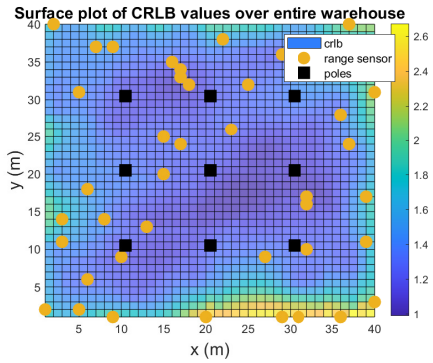
In all the performed tests below, the acceptable CRLB limit for all possible grid positions was set to 9m^2. This equals a standard deviation of 3m. In this section, the resulting CRLB values will be discussed by the standard deviations. The reason for this is that expressing the obtainable precision in meters instead of meters squared creates more understandable values.

5.2.1 Optimization of Sensors Placements

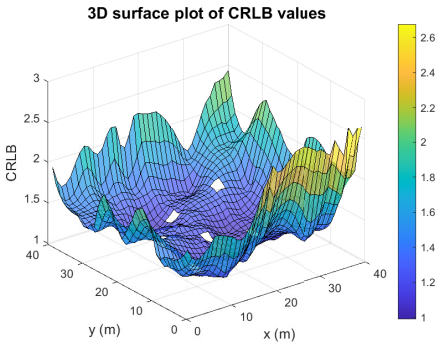
Range estimating sensors only

The first optimization was performed by only placing sensors that measured range. The optimization was performed 5 different times, resulting in different values for the resulting number of sensors needed. The initial number of sensors was set to 38, which was decided by doing some tests with random numbers and choosing the number which was too low to provide feasible answers. After the 5 iterations, 3 out of 5 optimal solutions required 40 range sensors to provide sufficient coverage and accuracy. The resulting number of sensors from the different runs are shown in Table 5.2.

Figure 5.8 shows the resulting CRLB values for one of the results which required 40 ranging sensors. Big parts of the grid have a dark purple color, meaning a really low CRLB value. However, the edges have some spikes, causing the high number of sensors needed.

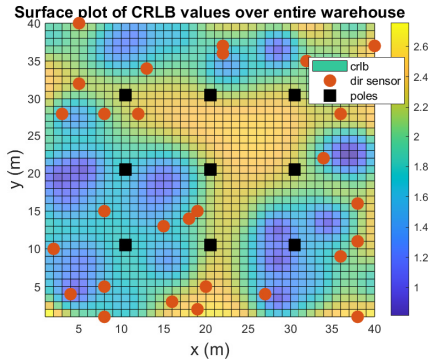


(a) 2D plot showing sensor placements and each grid colored wrt. the CRLB value.

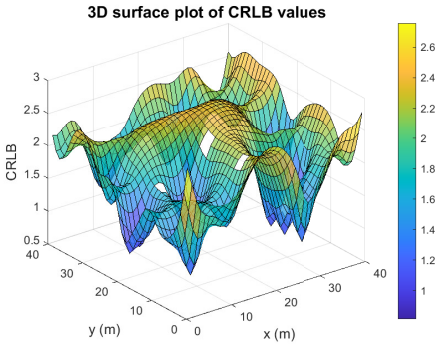


(b) 3D surface plot of CRLB in the grid when only using range sensors.

Figure 5.8: Illustration of warehouse and sensor placements with corresponding CRLB values with only range sensors. 40 range sensors used.



(a) 2D plot showing sensor placements and each grid colored wrt. the CRLB value.



(b) 3D surface plot of CRLB in the grid when only using direction sensors.

Figure 5.9: Illustration of warehouse and sensor placements with corresponding CRLB values with only direction sensors. 28 direction sensors were used.

Direction Finding sensors only

The tests involving only Direction Finding devices were conducted in a similar manner to the tests involving only range beacons. The initial number of sensors was set to 28, as fewer direction sensors appeared to be needed to cover the whole area after performing some random runs. After the 5 optimization runs, the number of sensors needed varied from 28 to 30. The different results from the runs are shown in Table 5.2.

Two of the results agreed that it was sufficient with 28 direction measuring sensors, and Figure 5.9 shows one of these cases. By comparing this case with the previous, where only range sensors were used, there are a lot fewer sensors needed to provide sufficient accuracy. However, by looking at the plots, the overall accuracy appears to be better when the high number of ranging sensors were used.

Table 5.2: Resulting number of sensors in the two separate optimizations.

Run	1	2	3	4	5
Range only	41	42	40	40	40
Direction only	28	28	30	30	29

Combination of range and Direction Finding sensors

When both range beacons and Direction Finding devices were experimented with together, it was assumed that it existed more possible configurations of sensor placements than in the previous tests. For this reason, the optimization problem was performed 10 times to accurately determine the best number of sensors.

As the two previous tests confirmed that there were fewer direction sensors than range sensors needed in the grid, the initial guess for the number of sensors, in this case, was 9 range sensors and 11 direction sensors. Choosing more direction sensors may result in fewer sensors needed overall. After 10 iterations, the resulting number of sensors needed from each run is shown in Table 5.3. The average total number of sensors needed resulted in 23.2, while the lowest possible with only one measurement type was 28.

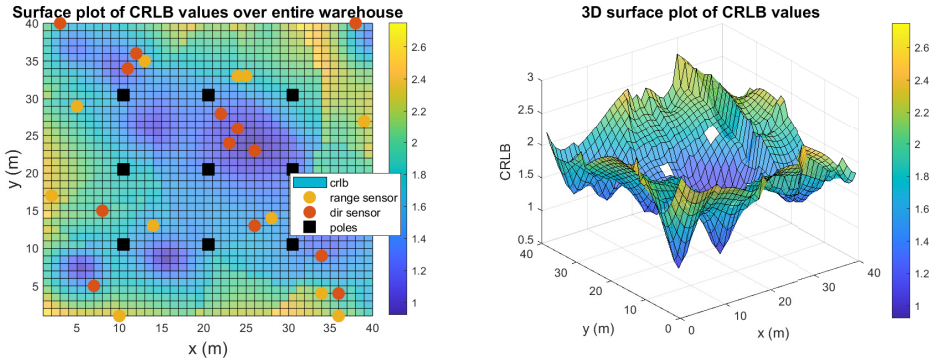
The average number of sensors, after rounding the numbers up, was 11 range sensors and 13 direction sensors. Figure 5.10 illustrates the results from run 1, which was the optimization run with the lowest CRLB peak and the same number of sensors as the average. From the colors in the plots, it can be seen that there are some small peaks spread over the grid and not one peak that is significantly larger than the others. Figure 5.10a shows that some of the sensors are clustered together.

From Table 5.3, it can also be deducted that the run with the fewest sensors is not necessarily the case with the worst CRLB peak. Run 5 was among the ones with the lowest number of sensors, but has a lower peak and average standard deviation than some of the cases where a higher number of sensors were used. Run 5 is illustrated in Figure 5.11. It can be seen that there are more uncertainties toward the walls in the room compared to Figure 5.10. However, the peaks remain under 3m which is within the accuracy limit. This case may be applicable as it uses the least amount of sensors in total. As Figure 5.11a displays, the sensors appear to be less clustered than in Figure 5.10a.

In Table 5.3 the average CRLB standard deviation of the entire grid is also included. It can be seen that the average value is much lower than the peaks for all cases. None of the averages are over 2m, even though the limit is set to 3m. This shows that the peaks represent only a small part of the uncertainties and that most points on the grid have a much lower CRLB value.

Table 5.3: Resulting number of sensors when optimizing with both types available. CRLB values are written as the resulting standard deviations.

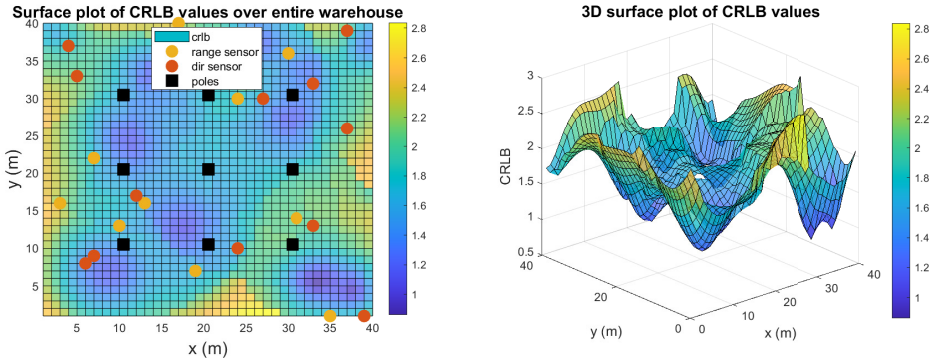
Run	Range sensors	Direction sensors	Total	CRLB peak	Avg. CRLB
1	11	13	24	2.75m	1.76m
2	11	13	24	2.92m	1.76m
3	10	13	23	2.97m	1.85m
4	11	13	24	2.93m	1.79m
5	10	12	22	2.84m	1.81m
6	11	13	24	2.85m	1.81m
7	10	12	22	2.97m	1.78m
8	10	13	23	2.95m	1.82m
9	11	12	23	2.75m	1.76m
10	10	13	23	2.96m	1.80m
Average	10.5	12.7	23.2		



(a) 2D plot showing sensor placements and each grid colored wrt. the CRLB value.

(b) 3D surface plot of CRLB when using both direction and range sensors.

Figure 5.10: Illustration of warehouse and sensor placements with corresponding CRLB values. Run 1. Average number of sensors with lowest CRLB peak.



(a) 2D plot showing sensor placements and each grid colored wrt. the CRLB value.

(b) 3D surface plot of CRLB when using both direction and range sensors.

Figure 5.11: Illustration of warehouse and sensor placements with corresponding CRLB values. Run 5. Fewest possible sensors with the lowest peak.

Reducing the Level of Accuracy to 90% and 95%

The last experiments were performed by removing the constraint that all the points in the grid must have CRLB standard deviation below 3m. Instead, the number of points with accuracy over the limit was counted and used to calculate the percentage of how many of the points had accuracy outside the constraints. There were two different percentages set as the limit, one being 90% and the other 95%. The 90% limit meant that the optimization problem accepted the sensor placements if less than 10% of the points had bad accuracy. Similarly for 95%, only with a stricter limit.

The optimization problem was executed 5 times for each of the percentages, and the resulting number of sensors in the 90% case are shown in Table 5.4. All the runs resulted in the same number of sensors, even though the initial sensors were 5 range sensors and 7 direction sensors. The results from the 95% limitation are shown in Table 5.5. It displays some variations in the number of sensors, but on average 8 range sensors and 10 direction sensors. The CRLB peaks from each run are also shown in the table. It can be seen that the peaks are higher for the cases of the 90% limit than the 95% limit. In addition, it is clear that even though all the optimization solutions for the 90% case used the same amount of sensors, the accuracy in the grids varied greatly, from 4.14m as the lowest to 8.56m as the greatest. The best and worst cases from the 90% limit tests are shown side by side in Figure 5.12. Figure 5.12a illustrates the run with the lowest peak, and it can be seen that the points with CRLB over 3m are spread over the whole grid. The results from run 2 with the highest peak show that almost all the CRLB values are gathered in one corner, as seen in Figure 5.12b.

The CRLB peaks from limiting to 95% accuracy are in general lower than the 90% limit peaks. It can be seen from Table 5.5 that the peaks are generally still quite close to the CRLB limit of 3m. The lowest peak is obtained in run 4 and shown in Figure 5.13a, and the worst peak is from run 2 and shown in Figure 5.13b. As in the 90% case, the best case has the high CRLB more spread over the entire grid, while the worst is gathered in one corner. However, the worst case with a 95% limit is still quite a lot better than the worst case in the 90% case.

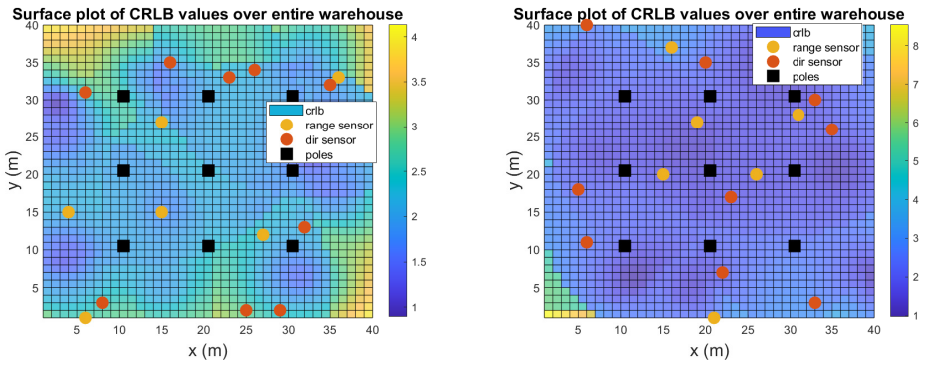
The calculated average values in both cases show that the standard deviation remains below 3m even for the runs with the worst peaks. The average values for the 95% case are in general lower than the average deviation in the 90% case, but the difference is low. From the 90% case, Table 5.4, it can be seen that the run with the highest peak actually has the lowest standard deviation on average. Additionally, in the 95% case, Table 5.5, the run with the highest peak has the second to best average standard deviation.

Table 5.4: Resulting number of sensors for the 90% percentage limit of accuracy. s. is short for sensors, Avg. short for average. The CRLB values are written as the resulting standard deviations.

		90%		
Runs	Range s.	Dir s.	CRLB peak	CRLB Avg.
1	6	9	7.92m	2.48m
2	6	9	8.56m	2.30m
3	6	9	5.11m	2.31m
4	6	9	4.92m	2.38m
5	6	9	4.14m	2.31m
Average	6	9		

Table 5.5: Resulting number of sensors for the 95% percentage limit of accuracy. s. is short for sensors, Avg. short for average. The CRLB values are written as the resulting standard deviations.

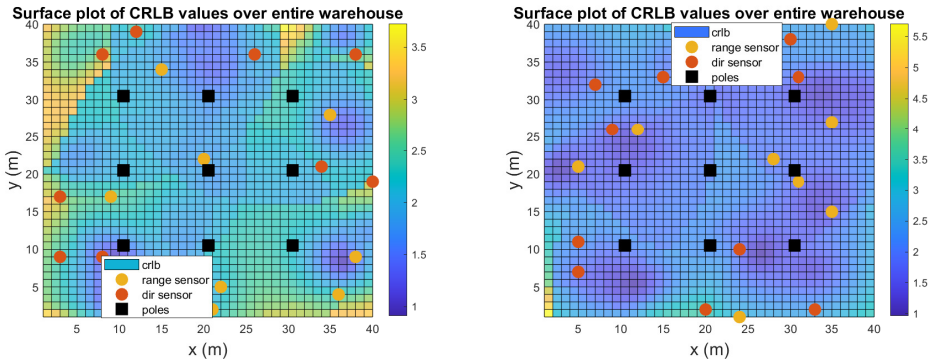
		95%		
Runs	Range s.	Dir s.	CRLB peak	CRLB Avg.
1	8	10	4.39m	2.17m
2	8	10	5.80m	2.15m
3	7	10	4.89m	2.17m
4	8	9	3.73m	2.17m
5	7	10	3.77m	2.09m
Average	7.6	9.8		



(a) Run 5. Lowest peak.

(b) Run 2. Highest peak.

Figure 5.12: 2D plots of CRLB values from the best and worst peak when using the 90% limit.



(a) Run 4. Lowest peak.

(b) Run 2. Highest peak.

Figure 5.13: 2D plots of CRLB values from the best and worst peak when using the 95% limit.

Chapter 6

Discussion

To examine the use of Bluetooth in navigation and localization cases, two different scenarios have been assessed. Both scenarios needed to be tested in simulated environments. As a result, the noises and expected errors were simulated as well. This is an important aspect that must be taken into account when evaluating the results. Errors which may be present in a fully operative scenario are hard to imitate and prepare for in a simulated scenario. This causes these simulated results to most probably deviate from expected results in a real test. However, the experiments were designed to include known issues and errors, to be as prepared as possible for how such systems will behave in real-world scenarios. Even though the applied estimators were subject to assumptions, the main focus of the thesis was to analyze how the sensor positions and resulting sensor geometry can be optimized to lower the uncertainty. The issue of perfecting the estimators may be a subject of interest after the optimal sensor placements have been found.

6.1 Discussion of the Net-Landing Results

The reason for performing the net-landing experiments was to show that Bluetooth Direction Finding can be used in navigation outdoors if aided by other measurements. This was done by showing that it is possible to provide robust navigation by including additional range-measuring beacons that have optimized placements. The problem was solved by describing the system using process and measurement models which could be used in an EKF. This allowed for the estimation of how the Bluetooth measurements were able to navigate in relation to a true trajectory.

During the development of the models, simplifications were made. One of the simplifications was ignoring orientation as a state of the UAV. The absence of information about the system's attitude could potentially have an impact on the navigation. This also resulted in the absence of rate-gyro in the IMU. Including a gyro would provide additional measurement in the control input of the process model which would contribute to better internal navigation, possibly making the system better at handling missing measurements. On the other hand, the gyro measurements would include additional errors in the process model, potentially

increasing the position covariance via the variance, cf. Section 4.1.2. However, as the main objective was to optimize the sensor positions to obtain the best accuracy, ensuring that the estimator was optimized was not the main focus.

The choice of using an EKF for the estimation was due to the nonlinear nature of the measurements with respect to the UAV's state. However, the EKF uses linearization and performs the estimation on the linearized system. This works great given mildly linear models, but the equations used in this case are quite nonlinear. The linearized system may therefore provide results that deviate from the true system, and how accurate the linearization is depends on the choice of the operating point. The system appeared very sensitive to the initialization values, and a reason for this may be the linearization. With good initialized matrices such as $\mathbf{P}_0$, the system would be less sensitive to the linearization and, as a result, provide better estimation. Poorly tuned metrics may result in huge deviations from the true value. Given this weakness in EKF, a possibility may be to evaluate different estimators. One technique is to use the Unscented Kalman Filter (UKF). While the EKF only uses one point in its approximation, the UKF uses multiple points called sigma points. These sigma points are deterministically chosen and represent the whole distribution, which results in more precise approximations. UKF is a method that overcomes the limitations that come from linearization in the EKF [54].

One of the assumptions which resulted in the use of an EKF was that the noises were Gaussian noise with constant variance. As briefly mentioned in the theory of Direction Finding, Section 2.1.2, the factors that affect the elevation angle error create a varying error. This will result in time-varying noises and consequently a varying measurement noise covariance matrix. Assuming that the noises are constant will therefore result in some errors when performing these experiments in the real world. It can therefore be interesting to look into methods for handling varying errors, such as adaptive filters.

The first test was performed only using the antenna array for navigation. The antenna array was a Bluetooth Direction Finding device that measured the azimuth and elevation directions. These angles were transformed into bearing measurements, describing the relative direction of the UAV with respect to the array. The estimation results were better than expected. It could be seen from the results in Figure 5.1 that the filter struggled with the estimation in the north direction, as the system had no measurements describing how far away UAV was from the array. The only information deducted from the antenna array was the bearing unit vector representing where the UAV is on a line. Since the antenna array axis was aligned with the NED axis, the measured azimuth explained the east angle while the elevation represented the down angle. As a result, the position estimation in these two directions was quite good as they were able to use both the IMU and direction measurements in the estimation. An increasing error was present in the north position as the direction measurements do not contain information about this and the IMU measurements are not stable enough to solely handle the navigation. The position error in the north direction in Figure 5.1a illustrates the

weakness of only relying on the antenna array for incoming measurements.

As the simulated scenario takes place over a relatively short time, the estimation accuracy stays within reasonable values. However, the error covariance matrix values show an increasing trend in the north position. Envisioning a longer simulation time, the error can be expected to increase a lot more. Consequently, it can be assumed that the error in the north direction will continue to increase as there are no additional measurements to correct for this drift. The result of this consequence is that navigation solely based on the direction will not provide sufficient accuracy for the UAV to land.

In addition to this, an aspect that is not covered in these tests is reflections on the water's surface. The reflection will lead to multipath, creating remarkable errors in the elevation angle. With only the direction and acceleration measurements, such errors would greatly affect the system as there are no other measurements to prevent these sudden changes. Since the elevation angle is present in the calculation of all the bearing values, this error will greatly affect the bearing and consequently the estimation. Considering these real-world issues, a navigation system only depending on the antenna array can not be considered robust.

The lack of full position observability in only using one antenna array was the reason for experimenting with additional beacons for range measurements. Even though it is a fact that BLE beacons measure ranges with a significant amount of errors, the experiments were performed to see if they can be used to assist the antenna array measurements. The geometry of range sensors has a huge influence on the accuracy of the object localization, which was a reason for performing an optimization based on moving the beacons around. The limited feasible range of the beacons was also a reason for exploring various sensor placements. Where the beacons are placed with respect to the trajectory would determine when the UAV is measured, hence affecting the accuracy throughout the trajectory. The accuracy was calculated by using the CRLB technique. This method was able to calculate where the beacons should be placed to provide good navigation of the UAV, either over the entire trajectory or close to the landing net. The resulting CRLB values illustrated how the bacon positions affected the accuracy of the system. The graphs resulting from the optimizations displayed that the system's standard deviations were increasing as long as no beacons were available. However, the moment a beacon was in the range of the UAV, the standard deviations had a sudden reduction. The deviations continued to decrease as more beacons got available. This illustrated the expected improvement in the estimation given additional beacons.

The optimization algorithm resulted in two different positions regarding where in the trajectory the optimization was focused on obtaining high accuracy. There were three different optimizations performed, however, two of the results were identical. Hence, only the cases focusing on good accuracy close to the net and good accuracy overall were considered. The resulting positions from both cases were able to achieve adequate uncertainty when the UAV was close to the net. The size of the net can be designed given its use case, hence the necessary accuracy close to the net will depend on the size. The choice of the best positions is

therefore also dependent on the wanted outcome. When the focus was only on assuring good accuracy close to the net, the uncertainty increased a lot throughout the trajectory before the beacons got available. In this case, the navigation further away will still be weak but the beacons strengthen the navigation toward the landing-net. In the case where the overall accuracy was a priority, the beacons were placed a bit further from the landing-net, ensuring coverage of the trajectory for a longer period, but with a reduced accuracy close to the net.

When the positions from the optimizations were used for estimation with the EKF, the actual estimation variance had much higher peaks than the CRLB value anticipated. The results also showed that the estimation got extremely unstable the moment range measurements were included in the model. However, as soon as more range measurements were available, the estimated states were able to move closer to the true value and stay close to it. There are different possible reasons for the sudden errors when the range measurements start. One reason may be that the system was already slightly inaccurate due to the linearization, hence the inclusion of extra input created an unbalance which resulted in erroneous estimation. Another reason could be that since the range measurements contained more noise than the direction and acceleration measurements, the inclusion of only one sensor provided more errors than assistance to the estimation. It can therefore be seen from the results that the big errors only occur when there were only one or two range sensors, while when all three range sensors were available the sudden changes in the estimation stopped and the result remained close to the actual state.

The CRLB variances are based on an ideal case, but as the results demonstrated, the actual variances obtained when performing the estimation were significantly larger. The changes in the variances throughout the estimation remained the same as the CRLB anticipated, but the values were almost doubled. The CRLB is a value given a perfect system, which means that since the results were far from optimal, it is possible to obtain better accuracies given more thorough tuning of the system. The initial covariance matrix $\mathbf{P}_0$ was obtained by tuning. When varying the values in this matrix it showed that the estimation was heavenly affected by the initialization. Prioritizing better tuning of this matrix may result in better results. The estimation may also benefit from tuning the noise covariance matrices $\mathbf{R}_k$ and $\mathbf{Q}_k$, regarding what factors it is advantageous that the system prioritizes.

The results from adding beacons to the model with the antenna array show no apparent improvement as the results appeared noisier; nonetheless, the additional measurements helped the estimation from deteriorating and may provide crucial compensation if the bearing measurements become heavily disturbed by multipath. Adding sensors elsewhere than by the antenna array may also help the system in case of blockage between the UAV and the antenna as there still may be sensors in the line of sight of the UAV. It shows from the results that the additional beacons also help the estimation of the position in the north direction. Relying only on the antenna array resulted in poor estimation in the north direction, demonstrating the need for additional measurements to avoid a drift that

can have critical consequences.

An interesting result that can be seen in all the different cases is the behavior of the estimated biases. Due to its implementation in the model, it has a decreasing value. However, the results show that the more the estimated positions and velocities deviated from the true values, the more the biases changed as well. In the case with only the antenna array measuring, the biases changed quite gradually. In the two cases where range measurements were used as well, the changes in the biases were much larger and happened more suddenly. The bias states were implemented to account for the biases in the system's INS, however, it also provides an additional metric in the estimation which tries to reduce the estimation errors. It can be seen that the changes happening when only using the antenna array were synchronized with the increasing error that could be seen in the estimation. The sudden changes of the biases when measuring ranges as well as directions started simultaneously as the range measurements started and the estimation error increased. As the estimator became more uncertain and the states attempted to stabilize the system, the bias estimations showed significant changes.

The results of this scenario highlight that relying solely on BLE Direction Finding may not offer sufficient support in a navigation case. However, incorporating multiple sensors in a well-determined geometry can significantly enhance the accuracy of a navigation system throughout a simulation. It is possible to move the beacons around in order to obtain the wanted coverage of a trajectory. Adding more than three range measurements will further improve the navigation accuracy, but also increase the cost and complexity. As proven by the results, there is a trade-off between wanted accuracy throughout the trajectory and wanted accuracy close, which is a choice that will change depending on the use case.

6.2 Discussion of the Warehouse Results

Even though there already have been performed multiple experiments with Direction Finding for indoor localization, the experiments performed here provide useful information about the usage of Direction Finding combined with BLE beacons in an industrial case where the accuracy is critical. The chosen scenario for this case was the localization of autonomous robots in a warehouse. The goal of determining how many of the different Bluetooth sensors were needed to achieve sufficient accuracy for all possible positions was solved by a randomized optimization algorithm calculating the CRLB for each point.

The randomization in the algorithm, Algorithm 3, may lead to a limitation in the credibility of the results. As the sensors were placed randomly and the only constraint was that enough sensors were needed to cover each point, there was no guarantee for the geometry of the sensors. As previously discussed, good geometry is crucial for achieving the best possible result, and since the implemented algorithm had no additional checks to ensure this, the randomized positions may suffer from poor geometry. This effect could be seen in many of the tests performed. There were in many cases multiple sensors clustered together

in the room, for example in Figure 5.8 and Figure 5.10, reducing the effectiveness of each sensor. An effect of this was that the algorithm may have concluded with needing more sensors to provide sufficient coverage of the entire grid, even though it would be enough if the sensors already implemented were placed with better geometry.

Executing each optimization multiple times was an attempt to avoid the worst cases of randomization. This appeared to work as expected. By inspecting the results of optimization using both direction and range measurements in Table 5.3, it can be seen that run 5 obtained the lowest peak with the fewest number of sensors among all the runs. This case is shown in Figure 5.11, which illustrates that the sensors were quite well spread. On the other hand, the results showed that another case, run 3, had a higher peak and average standard deviation, but required one more sensor to achieve this. This illustrates the importance of good geometry between the sensors. By having multiple runs with the same assumptions, the results would probably differ in both the number of sensors and achievable accuracy, and it will be possible to choose either the fewest number of sensors or the best accuracy depending on the use case. As the number of runs increased, it was expected that the uncertainty caused by randomization would decrease, leading to a higher level of confidence in identifying the chosen result as the best option.

Another option that could be implemented to minimize the effect of randomization was to increase the number of iterations performed before incrementing one of the sensors. By doing this, the runs would take much longer time to finish, but the algorithm would provide a more thorough search before concluding with the need for extra sensors. As this was solely an issue based on how time-demanding the optimization would be, the choice of the number of iterations and runs performed can be decided depending on how important it is to obtain a good accuracy with the fewest possible sensors and how much time one is willing to spend on optimizing. The most accurate optimization technique would be to test every possible combination of sensor placements and calculate CRLB for all cases and only return the case with the fewest sensors and lowest variance. This technique will take an extremely long time to optimize, but with enough time to spare, it is doable.

Unforeseen reflections and blockage of the Bluetooth signals have not been considered in these experiments, but are issues that are expected to be present in a real warehouse scenario. The inclusion of the poles in the warehouse was an illustration of how the sensors were affected by items blocking the signal, but it does not include any unexpected blockage. However, in such a warehouse case, there are few things that are expected to suddenly enter the room and block the signals. Reflections, on the other hand, are assumed to be a bigger problem. The assumed variances used to calculate the CRLB were based on previous indoor experiments where reflection was present, but cases where reflections cause larger and sudden errors were not taken into account. As the sensors and the grid are static, the issue with reflection may not cause a big issue in reality. The reason for this is that the reflection happening will always be the same as everything remains

in the same place, making it possible to calibrate the assumed errors. Errors such as reflection and blockage, which only affect the accuracy for a short period, are not critical issues in this scenario. The reason for this is that it only reduces the localization ability for a short period, which may slow down the efficiency, but not create any fatal mistakes. If the use case requires stable accuracy for all points at all times, a solution is to use additional sensors. As blockage is caused by something being in between a point and a sensor, additional sensors somewhere else in the room are unaffected by the blockage and are still able to provide measurements. This is also a valid solution to limit the reflection errors, as the errors get reduced by over-determining the localization system.

An issue that may affect the CRLB in the grid is that the sensors measure direction and range in 3D, while the location they measured is known to be in 2D. This may lead to an unnecessary error effect. As the grid is placed in $z = 0$ there is no need to include the resulting uncertainty of the height as the height is correctly known beforehand. Including a method for disregarding the height uncertainties may improve the accuracy of the sensors, hence reducing the number of sensors needed.

The optimizations with range and direction sensors were performed separately to strengthen the theory that Direction Finding is a more accurate localization technique than BLE beacons. The direction sensors have lower variance and higher range, resulting in fewer sensors needed to sufficiently cover the area. Even though Direction Finding sensors have such high accuracy, the results from optimizing with both sensors available resulted in fewer sensors needed in total than Direction Finding needed by itself. Choosing an excess of direction sensors over range sensors means that the direction sensors are most trusted to localize accurately, but the ranging sensors can aid the direction sensors.

The results from optimizing with both range and direction sensors displayed that even though many of the peaks were quite close to the CRLB limit, the average standard deviations over the whole grid were considerably lower. This indicates that the peaks only reflect a small area in the covered grid, while the rest of the grid is generally better covered. The extensive coverage in most areas suggests that there is an excess of sensors compared to what is actually required. A possibility can be to test if some sensors can be removed from the areas where the precision is greater than required and see if the coverage still remains good enough.

The final tests changed the accuracy constraint to only count for a certain percentage, which demonstrated that relaxing the accuracy constraint by a few percentages reduced the number of sensors quite a lot. When 90% of the points had high enough accuracy, the number of sensors was greatly reduced. The issue with allowing so many points to have bad accuracy is if all the points gather in the same area. The uncertainty in that area will then be terrible, and the system has no control there, resulting in high CRLB peaks. This was illustrated in Figure 5.12b, which has one corner with almost no sensors, hence worse coverage. The best case with the 90% turned out to be quite acceptable, however, there were still some larger areas with bad coverage as seen in Figure 5.12a. When instead choosing the

limit to be 95%, the number of sensors needed was still reduced quite a lot, while the peaks maintained a satisfactory level. Figure 5.13b showed the case with the highest peak. It only appeared to be a few points that were over the limit, while the rest were good. That may be enough to retain good knowledge over the point in the grid. The best case with a 95% limit, Figure 5.13a, illustrated some high values toward the edges and corners, but no large areas.

The average CRLB values of the different percentage tests, shown in Table 5.4 and Table 5.5, displayed an interesting behavior. Even though the peaks in some of the runs were far higher than the 3m limit, the average deviation still remained below the limit. The worst peak of the 90% tests was much higher than the others but resulted in the lowest average standard deviation. The reason for this is when all the badly covered areas are gathered in one corner, there are most likely no sensors in that area. As a consequence, the sensors need to be placed somewhere else, resulting in better coverage in the rest of the grid. Since all the 90% runs resulted in the same number of sensors, the difference in peak and average values were only dependent on the geometry of the sensors. The average values in the 95% limit were all close to 2m, which is a lot lower than the 3m limit. These results illustrate the possibilities for relaxing the accuracy constraint while still maintaining a good average accuracy. Both cases reduced the number of sensors needed compared to the case with strict accuracy. The choice of which percentage is preferable is dependent on the use case. The results show that it may be more uncertainty with a lower percentage, but the number of sensors needed was reduced a lot. On the other hand, the higher percentage did not reduce the number of sensors that much, but the peaks were not unrealistically big and the average standard deviation remained really low. Nevertheless, both cases illustrated the importance of good geometry of the sensor positions as the peak sizes changed a lot even though the number of sensors was mostly the same.

The lower bound of the variances was set to $9m^2$ for all the points in the warehouse. This number was randomly set, only illustrating a high accuracy overall. As the wanted accuracy is solely dependent on the use case and how critical good coverage is, these experiments only display that it is possible to place sensors in order to obtain the wanted accuracy. If a case depends on even higher accuracy, the same optimization algorithm can be executed, only changing the limit of acceptable CRLB values. This way it is possible to adjust the scenario either way necessary.

Based on the results from the various experiments it can be seen that using a combination of BLE Direction Finding and range beacons may provide sufficient coverage of a warehouse. Due to noises and limitations such as blocking and reflection, it may not be the best solution to apply Bluetooth devices on their own to handle the entire navigation. Nevertheless, as Bluetooth devices are cheap and easy to deploy, a navigation system utilizing Bluetooth may be a great solution as a spare system in addition to other existing systems. The Bluetooth system can be employed as a separate system and handle the navigation in case the others fail. Given the use of Bluetooth devices only as an emergency system, it may be suffi-

cient to limit the accuracy within a percentage, obtaining good enough accuracy in important sections while other sections can remain uncovered. This reduces the number of sensors needed, creating a particularly low-cost system. The cost of the current system is unknown, but it uses both cameras and other sensors which are expected to be quite more expensive than BLE devices. Investing in a cheap, but reasonably robust security system with BLE sensors may therefore be preferable.

Chapter 7

Closing remarks

7.1 Conclusion

The two different analyses conducted to examine the usage of Bluetooth sensors for navigational purposes provided useful insight. It is important to note that even though the tests aimed to include known errors for a realistic outcome, the simulated scenarios may deviate from fully operative industrial scenarios.

The UAV net-landing scenario was implemented to demonstrate how Bluetooth Direction Finding combined with additional range-measuring beacons can provide robust navigation outdoors. As it is desirable to keep the navigation system low-cost, it may appear suitable to only employ the antenna array for direction measurements. However, even though the range beacons appear to include more noise in the estimation, they provide valuable robustness to the system. The antenna array is prone to many errors which are not considered here, which makes it necessary to have more sensors to rely on during navigation to ensure robustness throughout. The different results from the optimization display the importance of thorough placements of the sensors.

The warehouse scenario illustrated how Bluetooth technology can be utilized for navigation indoors. The industrial case is an example of how Bluetooth sensors can be used even when there are high demands on the system's accuracy. The results showed the importance of sensor geometry, and how poor geometry may result in the need for additional sensors. Multiple runs of the optimization algorithm helped minimize the impact of the random sensor placements, allowing for the choice of fewer sensors or better accuracy depending on the use case. Factors such as unforeseen reflections were not considered in these analyses but are expected challenges in real warehouse scenarios. The results show that it is possible to utilize Bluetooth sensors indoors and that the number of sensors needed is only depending on how accurate the coverage is expected to be.

To conclude, Direction Finding is a more accurate navigation technique than Bluetooth beacons but it has been shown that using a combination of these sensors provides sufficient accuracy in both scenarios. The analyses provided insight into the limitations in both systems and how tuning and sensor placement optimization

is necessary to achieve accurate and robust results if tested in the real world.

7.2 Further Work

One main step to further develop the understanding of using Bluetooth technologies in navigation is to perform some experiments in the real world in order to collect data on how the sensors actually measure. With real measurements as a base, there is a possibility to create and experiment with more realistic scenarios, hence creating more reliable results. With the systems created here, it may not be enough information to directly deploy them in the real world. After experimenting with true data, testing the systems in real life can be possible.

As briefly mentioned in the discussion, an option to improve the net landing scenario is to implement the UKF. The EKF may not be the best-suited estimator in this case as it suffers from some limitations. It may therefore be beneficial to explore other estimators. Another option is to look into the benefits of adaptive filtering to consider the time-varying errors.

An improvement on the warehouse scenario can be to implement a more thorough optimization algorithm that can provide good sensor geometry and test different combinations of the number of direction versus range sensors. It may be beneficial to thoroughly optimize the placements by iterating over all possible combinations.

It will be interesting to see the possibilities that follow from Bluetooth if the SIG releases a high-accuracy ranging solution in addition to the existing Direction Finding feature. How the combination of high-accuracy Bluetooth range and Direction Finding features can be used to obtain accurate navigation systems is an evolution that will be fascinating to follow.

Appendix A

Additional formulas

The matrix $\mathbf{M}_\eta$ maps the measurement noise from spherical coordinates to Cartesian coordinates as shown in (A.1).

$$\boldsymbol{M}_\eta = \frac{\partial \boldsymbol{\eta}_a^r}{\partial \boldsymbol{\varepsilon}^r} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{pmatrix} \in \mathbb{R}_{2x3} \tag{A.1}$$

where:

$$\begin{aligned} m_{11} &= \frac{\partial \eta_{a,N}}{\partial \varepsilon_\psi^r} &&= \frac{-\eta_{a,E}}{b_\alpha \cdot b_\psi} \\ m_{12} &= \frac{\partial \eta_{a,N}}{\partial \varepsilon_\alpha^r} &&= \frac{\eta_{a,N} \cdot \eta_{a,D}}{\sqrt{\left({\eta_{a,N}}^2 + {\eta_{a,E}}^2\right)} \cdot b_\psi \cdot b_\alpha} \\ m_{21} &= \frac{\partial \eta_{a,E}}{\partial \varepsilon_\psi^r} &&= \frac{\eta_{a,N}}{b_\alpha \cdot b_\psi} \\ m_{22} &= \frac{\partial \eta_{a,E}}{\partial \varepsilon_\alpha^r} &&= \frac{\eta_{a,E} \cdot \eta_{a,D}}{\sqrt{\left({\eta_{a,N}}^2 + {\eta_{a,E}}^2\right)} \cdot b_\psi \cdot b_\alpha} \\ m_{31} &= \frac{\partial \eta_{a,D}}{\partial \varepsilon_\psi^r} &&= 0 \\ m_{32} &= \frac{\partial \eta_{a,D}}{\partial \varepsilon_\alpha^r} &&= \frac{-\sqrt{{\eta_{a,N}}^2 + {\eta_{a,E}}^2}}{b_\alpha} \end{aligned} \tag{A.2}$$

Bibliography

[1] J. Raquet and R. K. Martin, 'Non-gnss radio frequency navigation,' in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 5308–5311. DOI: 10.1109/ICASSP.2008.4518858.

[2] K. Gryte, T. H. Bryne and T. A. Johansen, 'Unmanned aircraft flight control aided by phased-array radio navigation,' *Journal of Field Robotics*, vol. 38, no. 4, pp. 532–551, 2021.

[3] M. Mondal, S. Shidlovskiy, D. Shashev and M. Khaddour, 'Non-gnss navigation systems in aerial vehicles (multi-rotors),' in *2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 451–455.

[4] C. Tsirmpas, A. Rompas, O. Fokou and D. Koutsouris, 'An indoor navigation system for visually impaired and elderly people based on radio frequency identification (rfid),' *Information Sciences*, vol. 320, pp. 288–305, 2015.

[5] J. Biswas and M. Veloso, 'Wifi localization and navigation for autonomous indoor mobile robots,' in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4379–4384. DOI: 10.1109/ROBOT.2010.5509842.

[6] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor and Z. Sahinoglu, 'Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks,' *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005. DOI: 10.1109/MSP.2005.1458289.

[7] G. Desjardins, *High-accuracy positioning performance with bluetooth technology*, Jan. 2022. [Online]. Available: https://www.bluetooth.com/blog/high-accuracy-positioning-performance-with-bluetooth-technology/ (visited on 27/04/2023).

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

[9] S. Löffler, F. Kroll, I. Becker and P. Hofstedt, 'Optimal beacon placement for indoor positioning using constraint programming,' in *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2022, pp. 1–8.

[10] Z. Jianyong, L. Haiyong, C. Zili and L. Zhaohui, 'Rssi based bluetooth low energy indoor positioning,' in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 526–533. DOI: 10.1109/IPIN.2014.7275525.

[11] R. Bembenik and K. Falcman, 'Ble indoor positioning system using rssi-based trilateration.,' *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 11, no. 3, pp. 50–69, 2020.

[12] P. Zand, J. Romme, J. Govers, F. Pasveer and G. Dolmans, 'A high-accuracy phase-based ranging solution with bluetooth low energy (ble),' in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–8. DOI: 10.1109/WCNC.2019.8885791.

[13] M. Woolley, 'Bluetooth core specification v5. 1,' in *Bluetooth*, 2019.

[14] G. Pau, F. Arena, Y. E. Gebremariam and I. You, 'Bluetooth 5.1: An analysis of direction finding capability for high-precision location services,' *Sensors*, vol. 21, no. 11, p. 3589, 2021.

[15] P. Andersson and L. Persson, *Evaluation of bluetooth 5.1 as an indoor positioning system*, 2020.

[16] F. A. Toasa, L. Tello-Oquendo, C. R. Peńafiel-Ojeda and G. Cuzco, 'Experimental demonstration for indoor localization based on aoa of bluetooth 5.1 using software defined radio,' in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1–4. DOI: 10.1109/CCNC49032.2021.9369638.

[17] M. L. M. Sollie, K. Gryte, T. H. Bryne and T. A. Johansen, 'Outdoor navigation using bluetooth angle-of-arrival measurements,' *IEEE Access*, vol. 10, pp. 88 012–88 033, 2022.

[18] P. Sambu and M. Won, 'An experimental study on direction finding of bluetooth 5.1: Indoor vs outdoor,' in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2022, pp. 1934–1939.

[19] P. Sukumar, *Using bluetooth as a location technology*. [Online]. Available: https://www.silabs.com/whitepapers/using-bluetooth-in-location-tracking-devices?source=Partner&detail=Bluetooth-SIG&cid=prt-sig-blu-032723 (visited on 24/04/2023).

[20] C. Li, J. Trogh, D. Plets, E. Tanghe, J. Hoebeke, E. De Poorter and W. Joseph, 'Crlb-based positioning performance of indoor hybrid aoa/rss/tof localization,' in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2019, pp. 1–6.

[21] R. Faragher and R. Harle, 'Location fingerprinting with bluetooth low energy beacons,' *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[22] N. Sokolova, *Lecture notes in ttk5 satellite-based navigation techniques and applications*, fall 2022.

[23] K. Gryte, T. H. Bryne, S. M. Albrektsen and T. A. Johansen, 'Field test results of gnss-denied inertial navigation aided by phased-array radio systems for uavs,' in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2019, pp. 1398–1406.

[24] D. Tang, D. Lu, B. Cai and J. Wang, 'Gnss localization propagation error estimation considering environmental conditions,' in *2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, 2018, pp. 1–7. DOI: 10.1109/ITST.2018.8566771.

[25] Y. H. Ho and H. C. Chan, 'Decentralized adaptive indoor positioning protocol using bluetooth low energy,' *Computer Communications*, vol. 159, pp. 231–244, 2020.

[26] S. He and S.-H. G. Chan, 'Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons,' *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2015.

[27] M. Woolley, 'Bluetooth direction finding,' *A technical Overview*, 2019.

[28] A. Schnell, 'Fusion of bluetooth and ins measurements for robust navigation with extended kalman filter,' Unpublished manuscript, 2022.

[29] B. SIG, *Specifications in development*. [Online]. Available: https://www.bluetooth.com/specifications/specifications-in-development/ (visited on 24/04/2023).

[30] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin and Y. Pang, 'Indoor positioning algorithm based on the improved rssi distance model,' *Sensors*, vol. 18, no. 9, p. 2820, 2018.

[31] B. SIG, *Answered: Five key questions about bluetooth location services*. [Online]. Available: https://www.bluetooth.com/blog/answered-five-key-questions-about-bluetooth-location-services/ (visited on 24/04/2023).

[32] R. B. Langley *et al.*, 'Dilution of precision,' *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[33] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[34] M. L. Sollie, T. H. Bryne and T. A. Johansen, 'Pose estimation of uavs based on ins aided by two independent low-cost gnss receivers,' in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1425–1435. DOI: 10.1109/ICUAS.2019.8797746.

[35] E. Brekke, 'Fundamentals of sensor fusion: Target tracking, navigation and slam,' *Teaching material for future course in sensor fusion, NTNU*, 2018.

[36] X. R. Li and Z. Zhao, 'Measures of performance for evaluation of estimators and filters,' in *Signal and Data Processing of Small Targets 2001*, SPIE, vol. 4473, 2001, pp. 530–541.

[37] K. Alexis, *Aerial robotic autonomy: Methods and systems. nonlinear non-gaussian estimation*. [Online]. Available: https://www.autonomousrobotslab.com/uploads/5/8/4/4/58449511/ara-deepdive-nnge.pdf (visited on 28/04/2023).

[38] A. Zhu, *Precision and accuracy: What is the difference?* 2023. [Online]. Available: https://medium.com/geekculture/precision-and-accuracy-what-is-the-difference-fdea8b437495 (visited on 25/05/2023).

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

[40] G. Hendeby, 'Fundamental estimation and detection limits in linear non-gaussian systems,' Ph.D. dissertation, Institutionen för systemteknik, 2005.

[41] S. Date, *An intuitive look at fisher information. what it means, and why it's calculated the way it is calculated*, 2021. [Online]. Available: https://towardsdatascience.com/an-intuitive-look-at-fisher-information-2720c40867d8 (visited on 02/03/2023).

[42] M. Viswanathan, *Score, fisher information and estimator sensitivity*, 2012. [Online]. Available: https://www.gaussianwaves.com/2012/10/score-and-fisher-information-estimator-sensitivity/ (visited on 25/05/2023).

[43] E. F. Brekke, *Fundamentals of probabilistic estimation - from probability spaces to pcrlb*, Jan. 2015.

[44] F. Gustafsson, *Statistical sensor fusion*. Studentlitteratur, 2010.

[45] NTNU Cybernetics, *UAVlab - Net landing experiments*, YouTube video, 2020. [Online]. Available: https://www.youtube.com/watch?v=n4XhzcKLgm8 (visited on 16/05/2023).

[46] MathWorks, *Minjerkpolytraj*, 2023. [Online]. Available: https://se.mathworks.com/help/uav/ref/minjerkpolytraj.html (visited on 14/04/2023).

[47] MathWorks, *Waypointtrajectory*, 2023. [Online]. Available: https://se.mathworks.com/help/nav/ref/waypointtrajectory-system-object.html (visited on 14/04/2023).

[48] E. Logic, *Autostore*. [Online]. Available: https://www.elementlogic.no/losninger-og-tjenester/autostore/ (visited on 21/04/2023).

[49] J. Solà, *Quaternion kinematics for the error-state kalman filter*, 2017. arXiv: 1711.02508 [cs.RO].

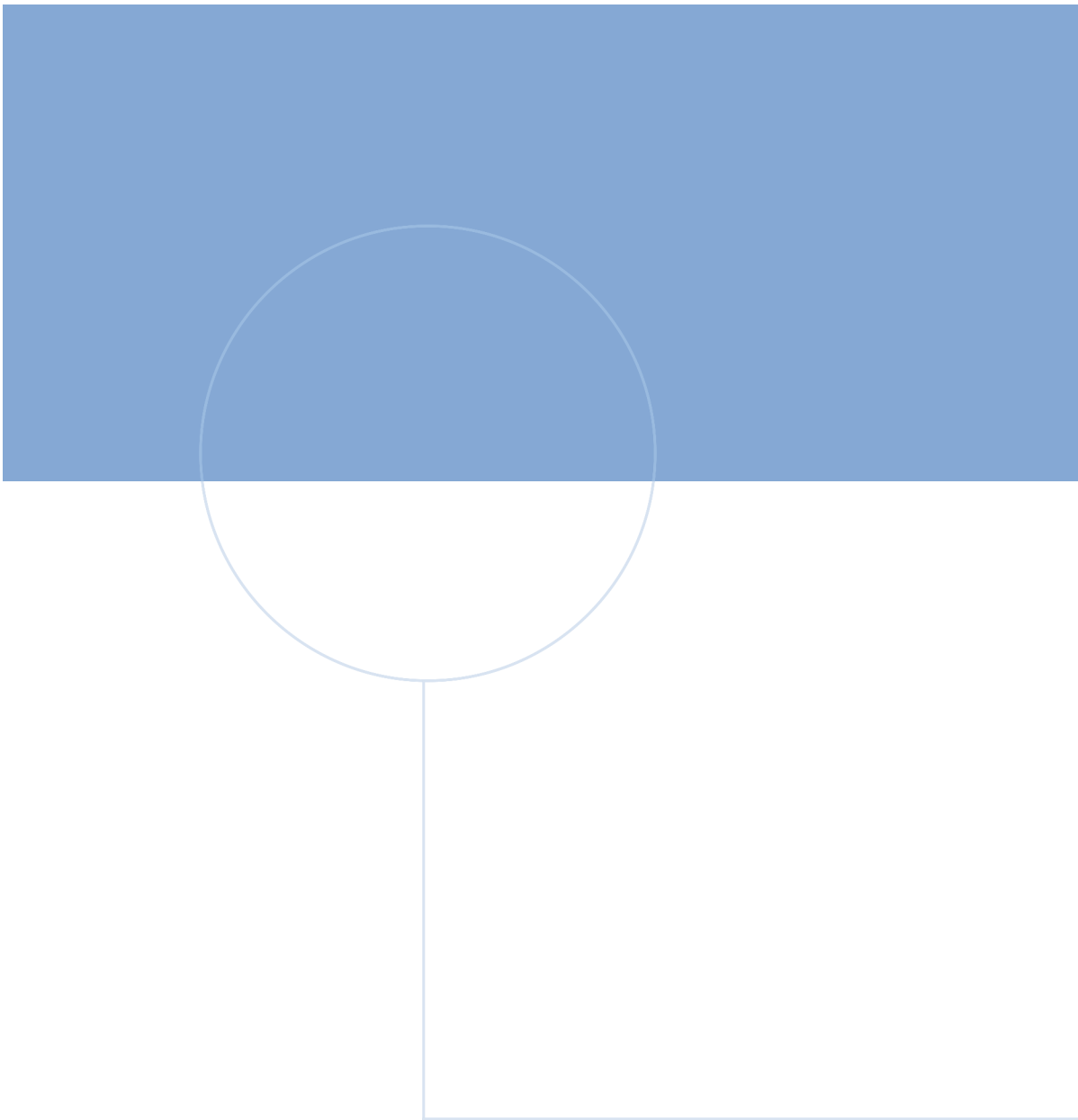
[50] S. Brorson, *Forward euler method*, 2021. [Online]. Available: https://math.libretexts.org/Bookshelves/Differential_Equations/Numerically_Solving_Ordinary_Differential_Equations_(Brorson)/01%3A_Chapters/1.02%3A_Forward_Euler_method (visited on 16/05/2023).

[51] O. Hasler, T. H. Bryne and T. A. Johansen, 'Robust phased array navigation in reflective prone areas,' in *Euopean Control Conference 2023 (ECC)*, 2023.

[52] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014, vol. 1286.

[53] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[54] S. Julier and J. Uhlmann, 'Unscented filtering and nonlinear estimation,' *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004. DOI: 10.1109/JPROC.2003.823141.



NTNU

Norwegian University of
Science and Technology