



HAMM-LIPPSTADT
UNIVERSITY OF APPLIED SCIENCES

Indoor 2D Tracker Based on UWB with Kalman Filtering

Department Lippstadt 2

Project Work

submitted by

Rubayet Kamal
Electronic Engineering
Mat.Nr.: 2219943
rubayet.kamal@stud.hshl.de

December 11, 2024

First supervisor: Prof. Dr. Stefan Henkler
Second supervisor: Lukas Walter

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Goals | 2 |
| 1.3 | Overview | 3 |
| 2 | Fundamentals | 5 |
| 2.1 | Parameters | 6 |
| 2.2 | Terminology for Radio Location | 6 |
| 2.3 | Positioning Technique | 6 |
| 2.4 | Geometric Principle | 8 |
| 2.5 | Kalman Filter | 9 |
| 2.6 | Overall Flow Chart | 11 |
| 3 | Related Work | 13 |
| 4 | Analysis | 15 |
| 4.1 | Hardware | 15 |
| 4.1.1 | Regulation | 15 |
| 4.1.2 | Makerfabs ESP32 UWB | 16 |
| 4.1.3 | Calibration | 16 |
| 4.2 | Specification Diagrams | 17 |
| 4.2.1 | Requirement Diagram | 17 |
| 4.2.2 | Use Case Diagram | 18 |
| 4.2.3 | Sequence Diagram | 18 |
| 4.2.4 | Block Definition Diagram | 19 |
| 4.2.5 | Activity Diagram | 20 |
| 5 | Design and Implementation | 21 |
| 5.1 | System Setup | 21 |
| 5.2 | Integration of Kalman Filter | 22 |
| 5.2.1 | Parameters | 22 |
| 5.2.2 | Kalman Filter Algorithm | 24 |
| 5.3 | Communication Protocol | 24 |
| 5.3.1 | SPI | 24 |
| 5.3.2 | I/O | 25 |
| 5.3.3 | TCP/IP | 25 |
| 6 | Evaluation | 27 |
| 6.1 | Stationary Tag | 27 |
| 6.1.1 | LOS | 27 |

| | | |
|-------------------|-----------------------------|-----------|
| 6.1.2 | NLOS | 27 |
| 6.2 | Moving Tag | 28 |
| 6.2.1 | LOS | 28 |
| 6.2.2 | NLOS | 29 |
| 6.3 | Improvisation | 29 |
| 6.4 | Self-localization | 30 |
| 7 | Outlook and Summary | 31 |
| 8 | Acknowledgements | 33 |
| References | | 34 |

1 Introduction

This project report details the design and implementation of an indoor location tracker via Ultra-wideband (UWB). DW1000 of Decawave, operated by ESP32, is used as base station and mobile tag. Positioning technique known as double sided Two-way Ranging (DS-TWR) combined with geometric principle known as Triangulation are implemented to calculate real-time coordinates of the vehicle. This project demonstrates the ability of UWB in indoor Line-of-sight (LOS) and No-Line-of-sight (NLOS) environment to locate and give precise real-time coordinates of the moving object. Additionally, the vehicle will know it's current location at all times allowing it to make decision based on the coordinate.

Section 1.1 discusses the motivation for choosing UWB as the optimal radio technology for indoor location tracking. The project goals, considering the specific environment and resources, are outlined in Section 1.2. Finally, Section 1.3 provides a brief roadmap of the project's implementation.

1.1 Motivation

In today's world, indoor location services have become an integral part of daily life. A prime example is industrial tracking, where precise indoor positioning of forklifts can optimize routes and improve operational efficiency. Additionally, geofencing, a technology that combines awareness of the user's current location with proximity to points of interest—is another application where real-time indoor tracking is critical, particularly in preventing collisions between machines or forklifts and people.

In September 2019, Apple introduced UWB (Ultra-Wideband) technology in its iPhone 11 lineup, including the iPhone 11, iPhone 11 Pro, and iPhone 11 Pro Max [Sne19]. This decision was driven by the unique capabilities of UWB in enabling precise location tracking and secure, short-range communication [Sne19]. UWB technology is highly accurate for determining the position of devices within a few centimeters, making it ideal for use cases like device-to-device spatial awareness, secure file sharing, and precise indoor navigation [Sne19]. "This is more relevant than ever, as new data indicates that more people want an item tracker than ever, now that Apple has made them appear legit" [Gri21]. UWB powers the Apple AirTag, a personal item tracker that leverages precise location tracking to help users find lost items with centimeter-level accuracy [Gri21]. A GPS can give information with precision to metre level whereas Ultrawide-band can be precise to cm level [Wor19].

Following Apple's lead, other major smartphone manufacturers, such as Samsung and Xiaomi, are actively integrating UWB technology into their flagship devices [Fir24]. This growing adoption highlights the potential of UWB technology in shaping the future of smart devices, enabling new use cases in augmented reality (AR), contactless payments, hands-free car access, and enhanced security protocols.

The inclusion of UWB in consumer electronics marks a significant shift towards more precise and context-aware applications, which are expected to see substantial growth in sectors like automotive, healthcare, and smart cities. The relevance of UWB extends beyond consumer devices to include industrial automation, robotics, and IoT systems, where accurate positioning and low-latency communication are critical. Thus, exploring UWB technology and its applications is essential for understanding its impact on both current and emerging technological landscapes.

1.2 Goals

In July 2024, a prototype of an autonomous vehicle was developed as part of the Prototyping and Systems Engineering course, as shown in fig. 1.1. This prototype was demonstrated on a racing track, illustrated in fig. 1.2. The main objective of this project is to remotely monitor the movement of the autonomous vehicle. The vehicle's 2D coordinates are visualized in real-time using Python's Matplotlib library.

To evaluate the accuracy of the DW1000 UWB sensors, various scenarios were tested, including Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions, with the vehicle both stationary and in motion. The vehicle is designed to be aware of its current coordinate at all times, enabling it to make decisions based on its position.

Given the potential for sensor readings to be affected by environmental noise, a Kalman Filter was implemented to enhance the accuracy of the positioning data. Additionally, a custom threshold algorithm was introduced to filter out significant spikes in the coordinates, ensuring that only reliable data is used for navigation.



Figure 1.1: Prototype of Autonomous Vehicle

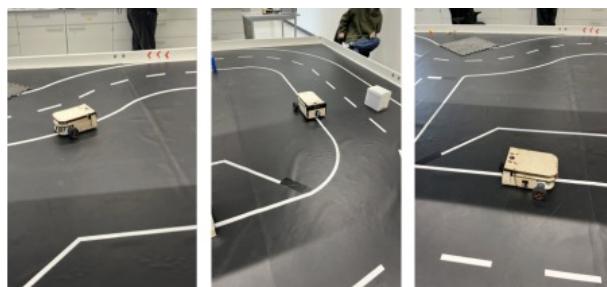


Figure 1.2: Car Following Line

1.3 Overview

Fundamentals of UWB will be discussed in Chapter 2. This will also include positioning technique, geometrical principle employed along with the estimation method called Kalman filter implemented in the project. In Chapter 3, a review of existent studies in the literature is presented. In Chapter 4, the Project has been analysed by discussing the regulation, hardware in use and specification diagrams using System Modelling Language. These diagrams allowed to display the implemented Algorithm and Flow Chart to illustrate the communication between two Anchors and one Tag. Based on this analysis as well as related work, an implementation is presented in Chapter 5 that shows how the scenario is set-up along with parameters of Kalman Filter that has been integrated. An evaluation of the developed approach is presented in Chapter 6 with matplotlib of Python. Here, the requirements set are reflected upon. Finally, a summary and outlook is presented in Chapter 7.

2 Fundamentals

"UWB communication systems are short-range radio communications that spread signal energy across a very large frequency range (FCC 2002; ITU-R 2006) [SDM14]."¹ As illustrated in fig. 2.1, the UWB frequency spectrum spans between 3.1 and 10.6 GHz. UWB, by definition, transmits information across a wide bandwidth of over 500 MHz [IKS21]. This allows UWB to transmit high levels of signal energy without causing interference with conventional narrowband or carrier wave transmissions in the same frequency range [IKS21]. This broader bandwidth enables UWB systems to achieve precise timing-based ranging, which is particularly valuable for accurate indoor location tracking [IKS21].

Moreover, UWB's excellent time resolution allows for the use of Time-of-Flight (TOF) techniques, facilitating the localization of mobile tags with centimeter-level precision [IKS21]. Given these advantages, UWB is ideally suited for asset tracking in high-density environments, as it remains relatively immune to multipath effects and interference from other devices [IKS21].

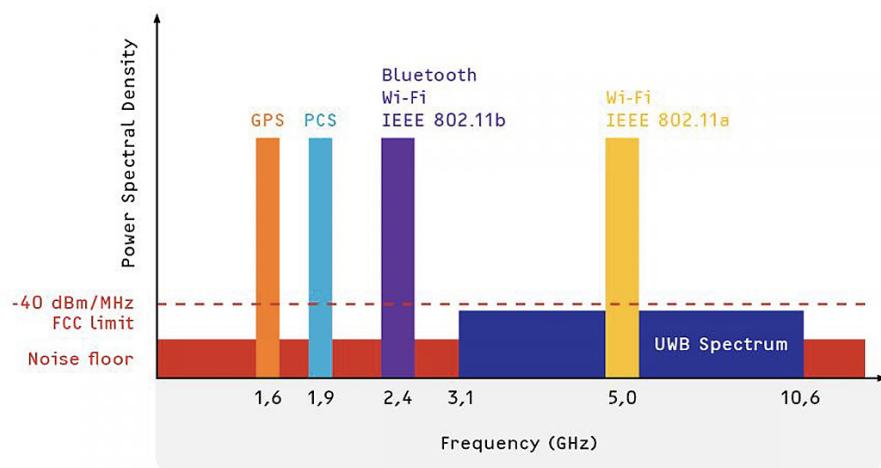


Figure 2.1: UWB frequency spectrum [IKS21]

The purpose of this chapter is to outline the fundamentals required for this project. Section 2.1 discusses the parameters which can have an effect on the readings received by the UWB sensors. Section 2.2 talks about the traditional terms used for radio location techniques. In section 2.3 & section 2.4, the method by which distance and coordinates are calculated are discussed. In section 2.5, the method to eliminate noise from the readings is highlighted. Finally, in section 2.6, a summary that shows the sequence of the execution of each step of the project is displayed.

2.1 Parameters

"Geometric parameters significantly influence the propagation of electromagnetic signals such as UWB:

- Propagation Delay: Electromagnetic Waves Travel With The Speed Of Light. This Yields Linear relation between the signal propagation delay and the distance of transmitter and receiver.
- Propagation attenuation: The electromagnetic signal power decreases as the distance between transmitter and receiver increases. In free space the received power is reciprocally proportional to the square of that distance.
- Reflection, refraction, diffraction, scattering: Discontinuities in the propagation medium cause changes in the propagation direction of electromagnetic waves. Such discontinuities are omnipresent in typical radio communications environments and cause NLOS and multipath propagation [SDM14]."

"In indoor environments with multi-path effects, reflections from nearby obstacles such as floors, ceilings, walls, and machinery can significantly affect the received signal. These reflections, combined with the direct signal, make it challenging to accurately estimate the distance between two RF transceivers, which relies on measuring the signal propagation time. As a result, distinguishing the direct path from the reflected signals is crucial for precise distance estimation [TSB17]."

2.2 Terminology for Radio Location

Radio location methods require a transmitter (TX) and receiver (RX) [SDM14]. DW1000 is a transceiver, therefore can act as both transmitter or receiver. Three DW1000 units—DW1000A, DW1000B, and DW1000C—are deployed. DW1000A and DW1000B will be stationary, commonly referred to as base stations. However, for the purposes of this paper, they will be called anchors, as they remain fixed throughout the operation. DW1000C, on the other hand, will be placed on top of the autonomous vehicle that will be moving and transmitting the initial signal. It is usually known as Mobile Terminal (MT) but here will be called as Tag.

Transmission of message from BS to MT is called downlink and vice versa is known as uplink [SDM14]. Both downlink and uplink of a mobile communications system has been considered in this project. The distance from each Anchor to the Tag will be calculated. The coordinate of the Mobile Tag will be calculated using the Geometric Principle discussed in Section 2.3 and Section 2.4. This real-time coordinate will then be displayed in Terminal of a computer from a Remote location.

2.3 Positioning Technique

Range estimation can be accomplished by obtaining the time of flight (TOF) of a signal between the tag and the anchor.

The Time of Arrival (TOA) algorithm assumes a common time base for all involved transponders [TSB17]. Transponder A transmits a message recording the timestamp of transmission, transponder B takes another timestamp on reception. Simple subtraction of RX/TX timestamps leads to the TOF, linearly concatenated to the distance over the propagation velocity:

$$TOF = t_{RX} - t_{TX} \quad (2.1)$$

$$d = TOF \cdot c \quad (2.2)$$

"The main drawback of Time of Arrival (TOA) in wireless sensor networks (WSN) is its reliance on a common time base. This requires the internal clocks of each module to be synchronized. Achieving such synchronization among all transponders is complex and resource-intensive, often requiring a wired synchronization channel or sophisticated wireless protocols. The Time Difference of Arrival (TDOA) method, which relies on time differences at multiple synchronized receivers, is also impractical for many applications. For instance, in a simplified 1D ranging system, TDOA cannot be used without at least two synchronized receivers. In contrast, the Two-Way Ranging (TWR) scheme offers a practical alternative for systems with unsynchronized transponders, as it eliminates the need for synchronization entirely [TSB17]."

In this project, (TWR) has been used as the method to determine the propagation time. "TWR is a circular positioning method that involves communication twice between the anchor and the tag to measure the distance, which is calculated using timestamps stored in the UWB modules at both ends. This approach avoids the need for synchronization, which entails an increase in complexity and cost. This technique requires bidirectional communication between the tag and each of the involved anchors. There are two common TWR schemes, called single sided TWR and double sided TWR (DS-TWR). Single sided TWR is only suitable for short distances. As DS-TWR has more packets exchanged between the Anchor and the Tag, a more accurate range estimation is possible . Also, unexpected clock shifts in UWB modules can cause errors in estimation. To mitigate the clock shift effect, the double-sided two-way ranging (DS-TWR) method is crucial, which requires the anchor and the tag to communicate for three times [SYG19]." Therefore, DS-TWR is used for this project.

As can be seen in fig. 2.2, the tag initiates the communication by sending the first message at time T_1 . The anchor records the timestamp T_1 from the tag's message upon reception. When the anchor receives the message, it logs the reception time as T_2 . After a certain period, at time T_3 , the anchor responds with a message that includes the timestamps T_2 and T_3 , corresponding to the reception time of the first message and the transmission time of the reply, respectively. Upon receiving this message, the tag records the timestamp T_4 . Subsequently, the tag transmits another message to the anchor at time T_5 , after a delay of $(T_5 - T_4)$. The anchor logs the reception of this message at time T_6 . Finally, after a slight processing delay, a packet containing all the timestamps is sent to the ESP32 attached to the anchor, which calculates the propagation time, t_{prop} , using the following formula 2.3:

$$T_{prop} = \frac{(T_4 - T_1)(T_6 - T_3) - (T_3 - T_2)(T_5 - T_4)}{T_5 - T_1} \quad (2.3)$$

Once T_{prop} is determined, the distance between the Tag and the anchor can be easily

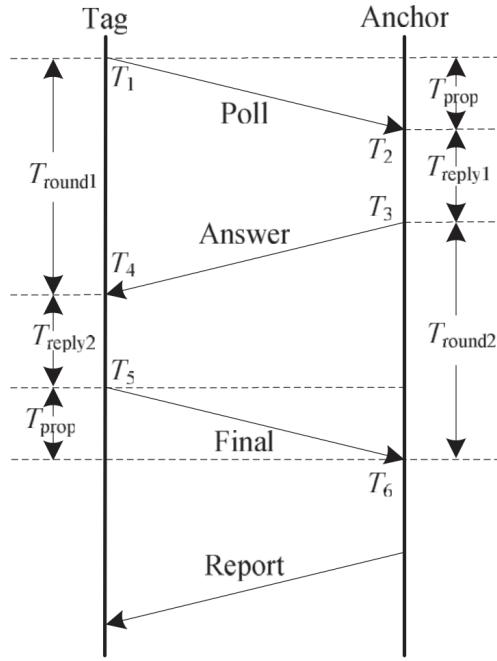


Figure 2.2: DS-TWR concept [SYG19]

calculated using the formula 2.4:

$$d = T_{prop} * C \quad (2.4)$$

where d is the distance between an anchor and a tag, $C = 3 \times 10^8 \text{ m/s}$ is the speed of electromagnetic wave propagation in the air. Once the distance between the tag and each anchor is calculated, the coordinate or location of the tag can be determined which is discussed in Section 2.4

2.4 Geometric Principle

This paper uses a positioning algorithm based on triangulation. Since the distance between the tag and each anchor has been obtained, the triangulation localization algorithm can be used to calculate the coordinates of the tag. Triangulation is the process of determining the location point of an object by measuring angles to the object's location from two or more beacons of known locations at either end of a fixed baseline, rather than measuring distances to the object's location point directly. The location point of the object can then be fixed as the third point of a triangle with one known side and two known angles. The triangulation principle is based on the laws of plane trigonometry, which state that, if one side and two angles of a triangle are known, the other two sides and angle can be readily calculated (Britannica 2009), and the location of a point is generally determined by measuring angles from beacons of known locations, and solving a triangle.

As shown in fig. 2.3 , we have three points. Two of these points, A and B, are the two anchors which are placed in fixed position. The distance between A and B is denoted by c which is known as it remains fixed. The tag is assumed to be the point C. The distance

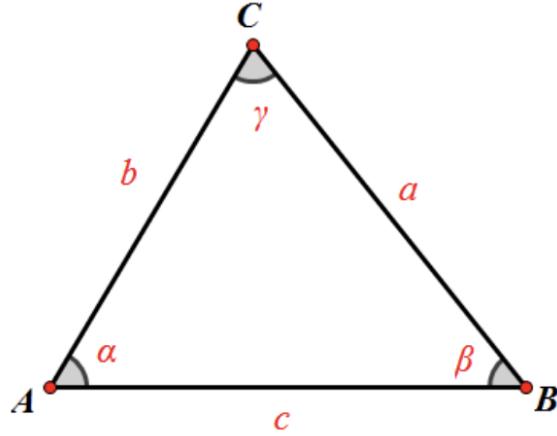


Figure 2.3: Triangle with all known sides

between the tag and each anchor is calculated already using DS-TWR discussed in Section 2.3. These two distances can be represented by b and a respectively. Therefore the length of all 3 sides of the triangle are known to us. It is to be assumed that all three UWB modules are at the same horizontal height. Point A has been set as the origin of the coordinate system. Now that we know the distance between the three sides of the triangle, we can calculate the coordinates of point "C" using:

$$\cos \alpha = \frac{b^2 + c^2 - a^2}{2bc} \quad (2.5)$$

As we know that,

$$\sin \theta = \sqrt{1 - \cos^2 \theta} \quad (2.6)$$

the coordinate of the Tag will be:

$$C(b \cos \alpha, b \sin \alpha) \quad (2.7)$$

2.5 Kalman Filter

"The Kalman filter is used to reduce measurement instability and noise errors" [IKS21]. "One of the main advantages of KFs is the computational efficiency in the implementation using only matrix and vector operations on the mean and covariances of Gaussian processes [SDM14]". Kalman filters are ideal for systems which are continuously changing. They have the advantage that they are light on memory (they don't need to keep any history other than the previous state), and they are very fast, making them well suited for real time problems.

The operation of Kalman Filter algorithm contains two steps: prediction step and estimation step. The prediction step is presented in the following two equations:

$$\begin{cases} \hat{x}_k^- = A\hat{x}_{k-1} \\ P_k^- = AP_{k-1}A^T + Q \end{cases} \quad (2.8)$$

where \hat{x}_k^- stands for mean coordinate prediction before taking the measurement, A is a $n \times n$ prediction matrix, P_k^- stands for error covariance (inaccuracy) of the prediction \hat{x}_k^- , Q is the $n \times n$ diagonal matrix which represents the process noise covariance, and k represents marching through time with t_k .

The next step is the estimation process where the dynamic Kalman gain K_k , the current estimate \hat{x}_k and the corresponding error covariance, P_k is calculated:

$$\begin{cases} K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \\ P_k = P_k^- - K_k H P_k^- \end{cases}$$

z_k is the measurement $m \times 1$ column vector and H is the state-to-measurement matrix $m \times n$. Initial coordinate of the Tag will be calculated using the algorithm and technique discussed in section 2.4 and section 2.3. This coordinate will be set as the initial value for \hat{x}_0^- while the later UWB-obtained results will serve as the measurement data, z_k . The initial data along with the initial error covariance will be used for the first prediction step. Result of the first prediction step will be used to calculate the best estimate after which the process will repeat itself. Because of Kalman filter, not only is the UWB random variation mitigated, but also a better heading direction estimation is obtained. "It should be noted that tweaking the values of Q and R are essential for optimal performance [WTMF⁺18]."

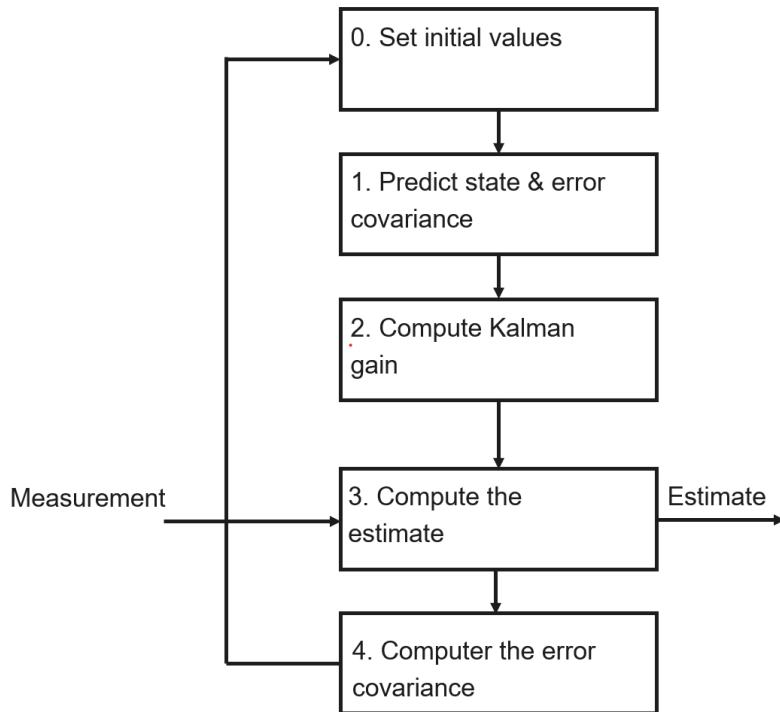
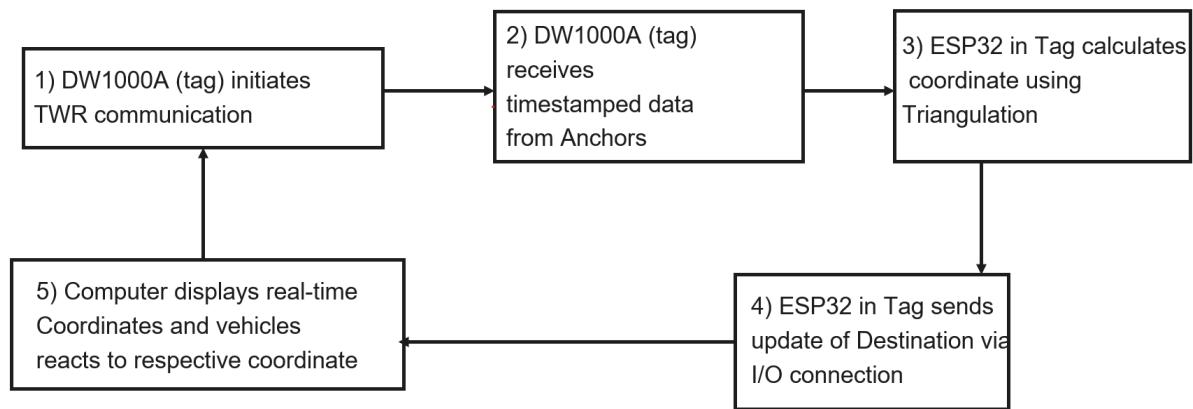


Figure 2.4: Kalman Filter Recursion

**Figure 2.5:** Overall Flow Chart

2.6 Overall Flow Chart

Fig 2.5 shows how the Tag and the Anchor communicate to calculate the coordinate of the moving Object. As shown below, first the Tag initializes the communication by sending a message to the Anchor. The Anchor responds according to the algorithm described in TWR in fig. 2.2. Once TWR is complete, both Anchor sends the calculated propagation time to the computer via Wifi. Using this data, the computer calculates the coordinate of the the Tag using the Positioning Technique discussed in section 2.4. Once calculation is complete, the process restarts with the Tag again sending initial message to both Anchors.

3 Related Work

Ultra-Wideband (UWB) technology has garnered significant attention in the field of indoor positioning due to its ability to provide high precision with low power consumption. Several research efforts have demonstrated the potential of UWB for centimeter-level accuracy, which is unmatched by other wireless technologies like Wi-Fi or Bluetooth, as shown in 2.1. Originally designed for military communication in the late 1960s [Bar00], it rapidly became popular for other application domains. Its peculiar characteristics allow for precise timing resolution, which is crucial to accurately estimate the Time of Flight (ToF) of the radio signal [RVdVSDP18]. Consequently, many researchers started to investigate this promising technology and its relevance in indoor positioning system. Immediately, important studies were published, showing that UWB was capable of localizing a node with a typical error of around 30 cm. Centimeter level accuracy is a significant performance improvement compared to other technologies such as WiFi and Bluetooth low energy (BLE), which can typically achieve localization accuracy in the order of a few meters [KY10].

Experiment in [ARR⁺07] led to the findings "that the largest target position estimation errors are located along the straight lines between TX and all RX antennas, and also on straight line between RX antennas, but in this case only behind antennas excluding area between them. The best system accuracy is obtained when distance between antennas is equal to $D = 5\text{m}$. The size of coverage area depends on arrangement of antennas and more RX antennas improve accuracy of UWB radar system." Furthermore, in [TSB17], the authors have calculated position estimates from unfiltered distance measurements acquired from time-of-flight (TOF) analysis. This raw data position estimation by the authors have concluded that there is better accuracy under LOS condition ($\pm 0.36 \text{ m}$) over obstructions blocking the LOS($\pm 0.93 \text{ m}$). In [VHVLVLDP24] the two algorithms proposed by the authors help to tackle the scalability and connectivity issue if there are too many anchors for tag as precious time is wasted trying to range with anchors that encounter poor channel characteristics. Anchors are selected in real-time based on their link quality, improving accuracy by selecting the best-performing anchor nodes. The algorithm's two variants are evaluated in a realistic dynamic industrial setup, reaching an accuracy up to 15 cm which represents a 50% improvement compared with ranging with all available anchors. The authors have claimed to be the first to evaluate both approaches with a real-time dynamic trajectory in an industrial environment and to compare them to reference scenarios.

In [RVdVSDP18], the author stated that Anchors must be synchronized accurately when using TdOA, which causes synchronization beacons overhead. However, for TWR, no synchronisation is necessary. Additionally, in [SXW⁺23], the authors have used DS-TWR for their experiment to mitigate the clock shift effect that takes place in traditional TWR. However, the author has claimed that inconsistency in height level between anchor and tag can lead to inaccuracy. In [LZB⁺21], the test results show that the positioning accuracy of UWB positioning system based on SDS-TWR algorithm is 90.83% higher than that

of the traditional two-way ranging (TWR) system, which verify the effectiveness of the proposed algorithm in the positioning system. In [BKH18], the authors claimed that several research has been performed on TDoA-UWB and TOA-UWB localization, the analysis about TWR-UWB localization especially in harsh environments like Underground mine is limited. Therefore, the authors performed a performance analysis for TWR-UWB localization in underground mine environments, where they showed the effectivity of UWB technology with TWR technique for harsh environment. Besides, the measurements have been performed in different channel frequencies, data rate and frame size. In LOS, the minimum error has been observed for L2 operational mode, which is associated with 2 channels, 4 GHz channel frequency and 110 kbps data rate. In NLOS, the minimum error rate has been observed for L5 operational mode, which is associated with 5 channels, 6.5 GHz channel frequency and 110 kbps data rate.

In [MX21], the author used Kalman Filter but with Trilateration as the Geometric Principle in order to overcome the errors in Non-Line-Of-Sight environment and improve the accuracy in both Line-Of-Sight and Non Line-Of-Sight scenarios. On the other hand, in [BCM⁺22], the author employed Extended Kalman Filter (EKF) for estimating the coordinates and angle in a complex system. Here in [BCM⁺22], inertial sensors were used to compensate for the limitation of UWB signals in NLOS or with barriers. Additionally, in [CYH21], the author used EKF in under-ground positioning system of coal mine. MATLAB was used to compare the initial error of TOA positioning mode and the positioning error of EKF TOA positioning mode. It has been concluded that EKF algorithm effectively improves the accuracy of UWB positioning system in LOS and NLOS conditions. In [BSY⁺23] paper, the authors presented a fusion of non-complex filtering algorithms which combines Kalman filter with Moving Average (MA) filter in order to reduce localization error using Ultra-Wideband (UWB). The performance of the technique was measured against the conventional method of Kalman filtering, and it was found that the average error was reduced even more by the proposed strategy compared to the standard Kalman filtering approach.

4 Analysis

This chapter dives into the primary hardware used in this project as well as the specification diagrams which have allowed for the analysis of the project. Section 4.1 discusses the regulation of frequency range by the European Union. This allowed to choose the hardware which follows the regulation. Additionally, the calibration used to set different hardwares were discussed. Section 4.2 presents the diagrams from System Modelling Language which allowed to analyse the project before execution.

4.1 Hardware

The purpose of this section is to outline the regulation allowed by the European Union which was crucial in deciding the hardware for the project. There this section is divided into three subsections: Section 4.1.1 talks about the frequency parameters allowed by the EU, section 4.1.2 discusses the chosen hardware & section 4.1.3 shows the steps taken to account for the variation in reading from the sensor.

4.1.1 Regulation

| Frequency range [GHz] | Maximum medium power spectral density height dBm/MHz (EIRP) | Maximum peak power related to 50 MHz |
|-----------------------|---|--------------------------------------|
| | | dBm (EIRP) |
| $f \leq 1.6$ | -90 | -50 |
| $1.6 \leq f \leq 2.7$ | -85 | -45 |
| $2.7 \leq f \leq 3.1$ | -70 | -36 |
| $3.1 \leq f \leq 3.4$ | -70 -41.3 with LDC/DAA | -36 0 |
| $3.4 \leq f \leq 3.8$ | -80 -41.3 with LDC/DAA | -40 0 |
| $3.8 \leq f \leq 4.8$ | -70 -41.3 with LDC/DAA | -30 0 |
| $4.8 \leq f \leq 6.0$ | -70 | -30 |
| $6.0 \leq f \leq 8.5$ | -41.3 | 0 |
| $8.5 \leq f \leq 9$ | -65 -41.3 with DAA | 25 0 |
| $9 \leq f \leq 10.6$ | -65 | -25 |
| $f \geq 10.6$ | -85 | -45 |

Table 4.1: Overview of frequency parameters according to [Bun16]

As is already discussed in Chapter 2, UWB systems use a very large bandwidth. Hence, interfering with conventional narrow-band radio systems in the same frequency domain is inevitable. As a conclusion in terms of high bandwidth of UWB, its related power spectral density (PSD) is very low compared to narrow-band systems [TSB17]. In the year 2007, the European commission already took a decision regarding the regulation of UWB systems. The decision should simplify the EU-wide operation of UWB-based products by harmonizing the country specific regulations [EU-07]. To fully comply with

the decision of the European commission, the Federal Network Agency for Germany (Bundesnetzagentur) published “Vfg. 37/2016” [Bun16] outlining the frame parameters for the use of UWB-systems in Germany. As per regulation, the greatest PSD without the use of additional mitigation techniques is permitted in the frequency band $6 \leq f \leq 8,5$ GHz. In this paper, this frequency band is used. While it fully complies to these German regulations, other local regulations may differ.

4.1.2 Makerfabs ESP32 UWB

To comply with the maximum allowed PSD (as shown in Table 4.1), the Makerfabs ESP32 UWB Development Board, which integrates the DecaWave DW1000 module, was selected. The DW1000 meets the required bandwidth of at least 500 MHz in the frequency range of $6 \leq f \leq 8,5$ GHz. The module is designed in accordance with the IEEE 802.15.4-2011 UWB PHY standard [IEE11]. The module supports both Two-Way Ranging (TWR) and Time Difference of Arrival (TDoA) systems, achieving location precision within 10 cm. Additionally, it offers data transfer rates up to 6.8 Mbps. Data transfer between the ESP32 and attached modules takes place through Serial Peripheral Interface (SPI). Time-stamping of outgoing and incoming messages is handled through an internal 64 GHz nominal counter (40-bit values), providing a minimal theoretical resolution defined by:

$$T = \frac{1}{f} = \frac{1}{64} = 15.65 \text{ ps} \quad (4.1)$$

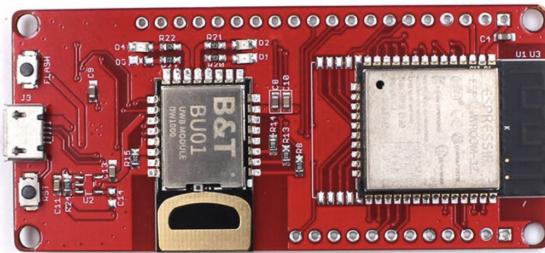


Figure 4.1: ESP32UWB

4.1.3 Calibration

DW1000 modules, like many other RF components, have small variations in their internal electronics due to manufacturing processes. This creates an offset in the measured distance, meaning that even if the actual distance between two devices is, for example, 1 meter, the measurement could show a slightly incorrect value. These offsets can vary from centimeters to meters. If not corrected, they would result in inaccurate positioning information, which is a problem for most applications. To correct this, a calibration is done. Two DW1000 are placed exactly 1, 2 and 3 meter apart, with the distance verified using a metre ruler. Using Two-Way Ranging (TWR) discussed in section 2.2, the estimated distance is measured by the system, and the difference between the measured value and the actual distance

is recorded as the offset. The system then compensates for this offset in its software processing, effectively "subtracting" the error from future distance measurements between these two transponders. While this calibration method works well for small-scale systems, it becomes impractical for larger systems. The calibration process must be repeated for all pairs of devices in the system, and as the number of devices grows, the number of pairs grows exponentially, significantly increasing the efforts to calibrate. Table 4.2 shows the distance calculated between the two modules using different number of Modules and different distances from each other.

| Module ID | Distance | 1m | 2m | 3m |
|----------------|----------|-------|------|------|
| | A | 0.90 | 0.75 | 0.95 |
| B | 0.98 | 0.20 | 0.55 | |
| Average Offset | 0.94 | 0.475 | 0.75 | |

Table 4.2: Ranging Offset

Table 4.2 shows the reading before the offset is applied. For example, with DW1000A as Anchor, the Tag shows a reading of 0.10 m at 1.0 m meaning that an offset of positive of 0.90 has to be applied to match the correct measurement. This way it has been found that the offset is very high for close range. For this project, instead of using the average offset value, different offset has been applied for different anchors. For example, 0.80 m is added to readings from DW1000A as offset. On the other hand, for readings from DW1000B, 1.10 is added. This results in a precise ranging and therefore results in precise coordinate calculation.

4.2 Specification Diagrams

Before diving into practical work, it became necessary to specify requirements, use cases and activities that take place in this project. For better visualisation, diagrams from System Modelling Language (SysML) were adopted. It supported the specification, analysis, design, verification and validation of the Project. Subsection 4.2.1 discusses and presents the primary requirements of the project. In addition, subsection 4.2.2, subsection 4.2.3, subsection 4.2.4 & subsection 4.2.5 represent use case diagram, sequence diagram, block definition diagram & activity diagram respectively.

4.2.1 Requirement Diagram

Requirement Diagram is a diagram that belongs to SysML. The requirements of the Project were established and visualized through this requirement diagram, shown in fig 4.2, which offers a graphical representation of the requirements. The project goal is to develop an indoor location tracker that can monitor movement of a prototype of an autonomous vehicle as shown in the requirement diagram with Id = 001. This requirement itself contain another requirement which is shown with Id = 002. By integrating these requirements,

the aim is for the tracker to monitor and display real time coordinates the autonomous vehicle.

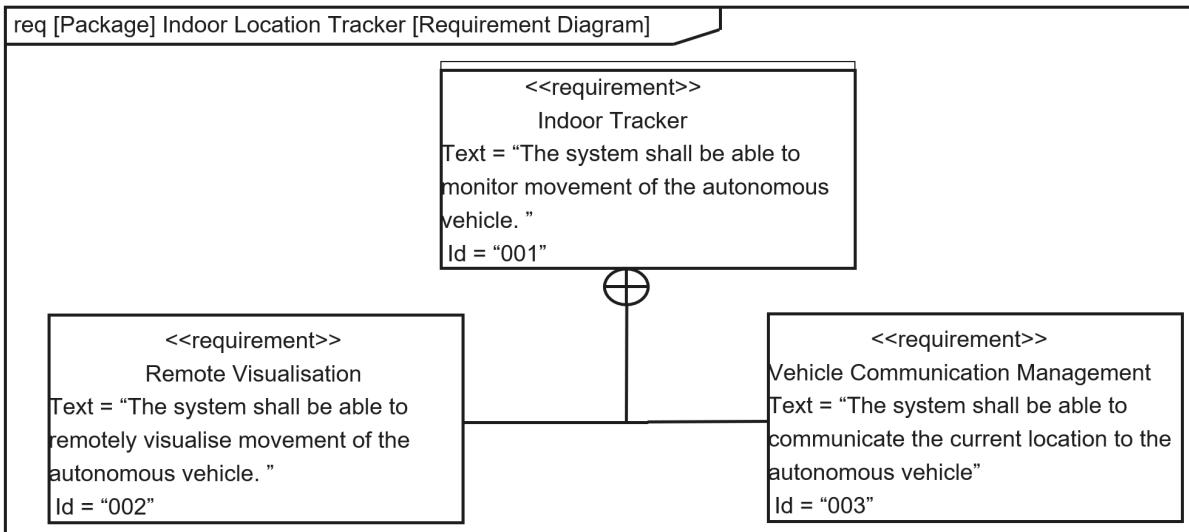


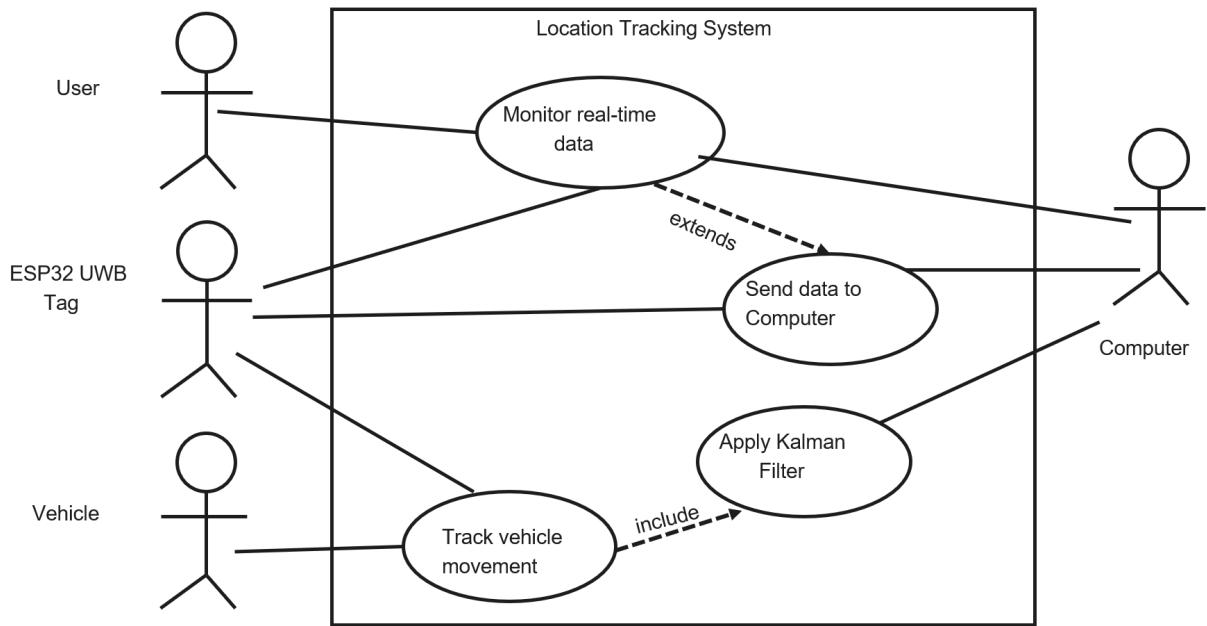
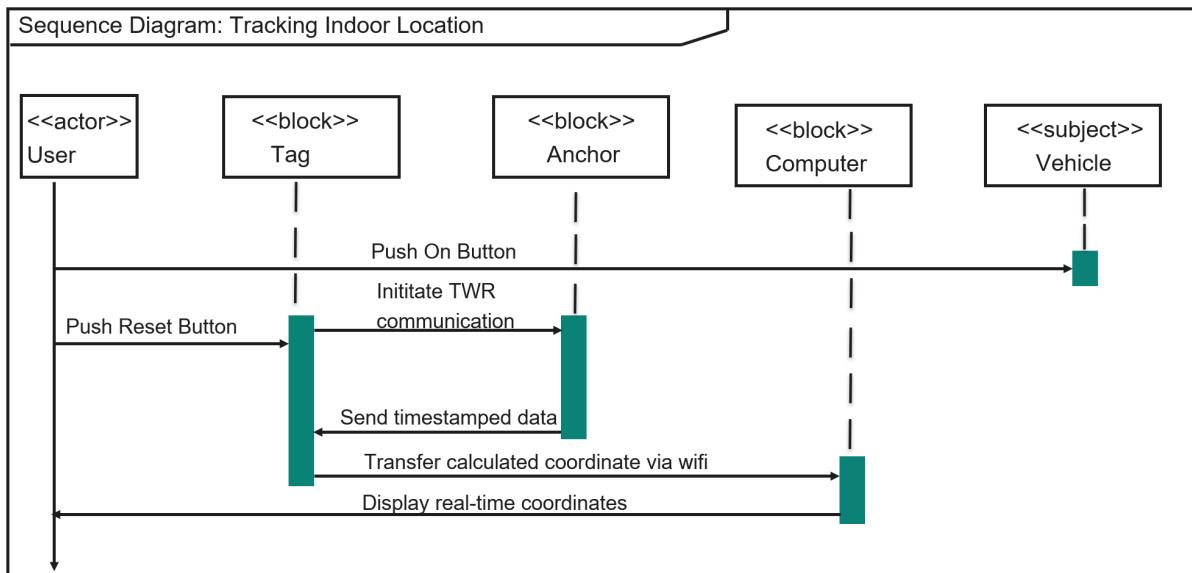
Figure 4.2: Requirement Diagram

4.2.2 Use Case Diagram

Displayed in fig. 4.3, the use case diagram for this system consists of four primary actors: the *User*, *Autonomous Vehicle*, *Tracking System*, and *Computer*. The **User**, which is me in this case, is responsible for monitoring the real-time coordinates of the vehicle and interacting with the system for data collection and analysis. The **Autonomous Vehicle**, on top of which the ESP32 UWB module is mounted, moves along the oval shaped track and continuously transmits its position. The **Tracking System**, which includes the ESP32UWB Tag and Anchors, collects and sends real-time location data of the vehicle. The **Computer** receives this data and displays the real-time coordinate of the vehicle using Matplotlib of Python with and without Kalman Filter. As seen in fig. 4.3 each actor plays a distinct role in the real-time tracking, monitoring, processing, and logging of vehicle movement.

4.2.3 Sequence Diagram

Fig 4.4 shows the order of communication between actors and sensors. The users starts the process by resetting the tag and by powering the vehicle on. As the tag completes the reset procedure, it initiates the TWR communication by sending a message to the Anchor. TWR is explained in section 2.2. The timestamped data is sent to the Tag by both Anchors to calculate the distance. Once the process of ranging is complete, the real-time coordinate calculated via Triangulation, explained in 2.3 is sent to the computer via Wifi which is displayed in terminal and graphed via matplotlib, a library of Python.

**Figure 4.3:** Use Case Diagram**Figure 4.4:** Sequence Diagram

4.2.4 Block Definition Diagram

Fig 4.5 shows the components used as blocks. This analysis helped in realising the relationship between different components. With the Tag, Anchors, Computer, Vehicle being part of this entire system, there is a Composition relationship between the system and individual base components. The relationship between the base components have been shown as Association with the the number representing multiplicity. For example, there is only instance of the Tag and two instances of Anchor, therefore the multiplicity is shown accordingly as one tag communicates with two anchors.

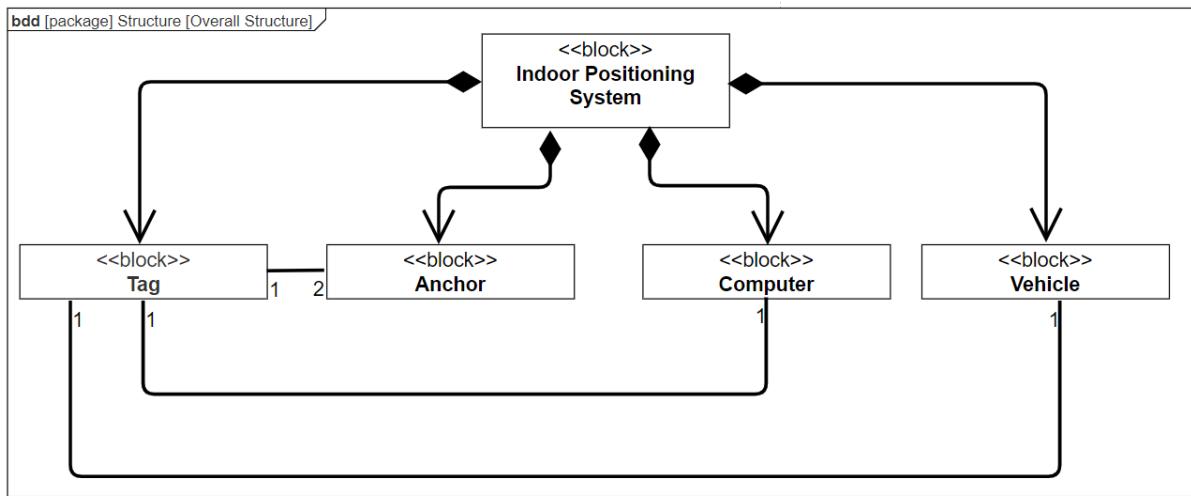


Figure 4.5: Block Definition Diagram

4.2.5 Activity Diagram

Fig 4.6 shows the actions that are taken by the system. As already discussed, it starts by pressing the reset button on the Tag, this way the tag initiates communication with anchors. As the task is to monitor movement of the vehicle on track, the vehicle is powered on at the same time. Once distance from each anchor is calculated and the Tag has all data in hand, the ESP32 attached to the Tag calculates the coordinate with respect to one of the Anchors as origin. Once coordinate calculation is complete, the message is passed to the computer via Wifi. At a zone, the vehicle is not allowed to pass. The vehicle is alerted about the point via Input Output connection. It is expected to perform a U-turn and start following the line in the opposite direction from before. The conditional symbol shows that the anchor keeps on communicating with the computer as long as it has power.

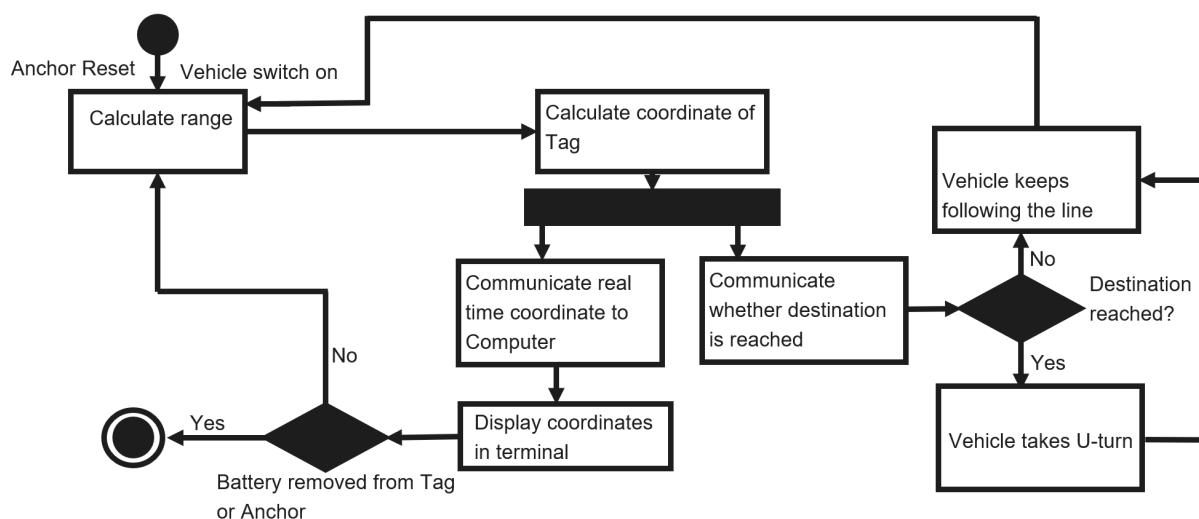


Figure 4.6: Activity Diagram

5 Design and Implementation

The purpose of this chapter is to present the setup and execution of the project as per the requirements and goal. Section 5.1 visualises the system in 2-D scenario, while discussing the pattern of movement of the Tag. Additionally, the placement of anchors in both LOS and NLOS scenarios will be shown. In section 5.2, the parameters of Kalman Filter used in this project will be discussed and the algorithm will be shown. Lastly, in section 5.3, the implementation of different methods used to cause communication between different systems and subsystems will be shown.

5.1 System Setup

Starting from the 1D-localization system which was used for calibration, the system was extended to two transponders as a stationary reference (anchor) and one mobile transponder (tags). The three anchor-tags were placed in a 2D- plane around the test area as shown in fig. 5.1. Estimating the tag position can be achieved by triangulation as discussed in Section 2.3. The distances are determined with TWR, the resulting set of data is transmitted over the UWB link to a central processing unit for further processing and visualization.

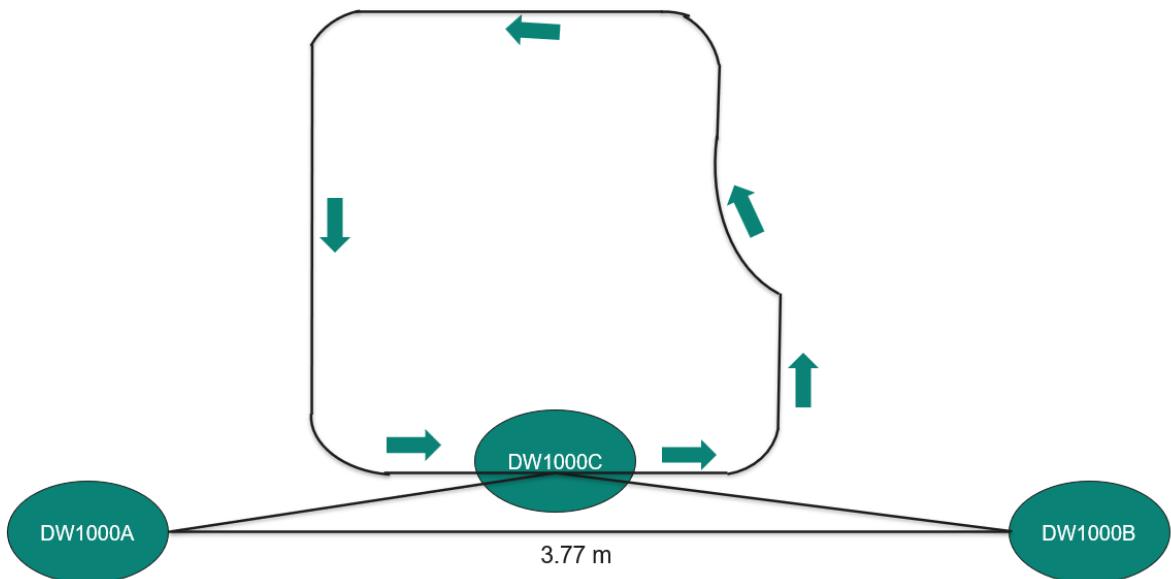


Figure 5.1: 2D Setup

The two figures below show the system has been set-up in lab. With two anchors being 3.77 m (calculated with tape measure) being away from each other, the coordinate of the Tag is monitored by placing it on top of the Autonomous Vehicle. The distance

between the two anchors is measured with a tape measure. Fig. 5.2 and fig. 5.3 shows how two different scenarios are set-up. The code uploaded to the Tag, Anchors, and the Autonomous Vehicle can be found at <https://github.com/RubayetKamal/UWB-Indoor-Tracker>. Additionally, the Python script for the Wi-Fi connection and remote monitoring of the vehicle is also available at the same repository. The DW1000 Library from https://github.com/jremington/UWB-Indoor-Localization_Arduino is used in this project.

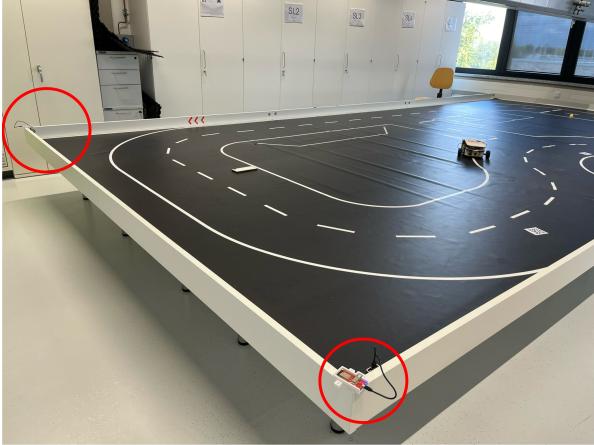


Figure 5.2: LOS

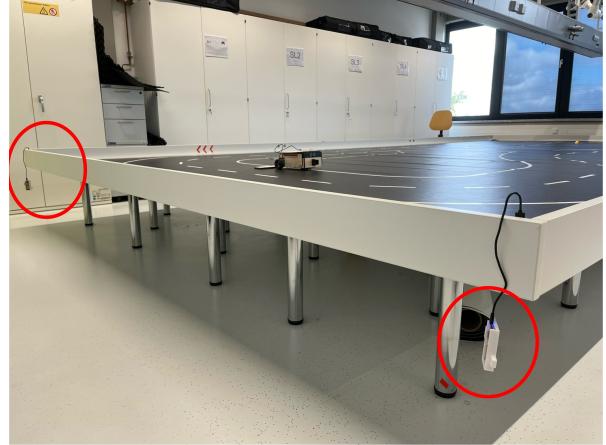


Figure 5.3: NLOS

5.2 Integration of Kalman Filter

Fundamentals of Kalman Filter has already been discussed in Section 2.5. This section is divided into two portions. In subsection 5.2.1, the parameters used while integrating Kalman Filter into the project is outlined. The parameters were written in Python that displays the readings using matplotlib. In subsection 5.2.2, the Kalman Filter algorithm is displayed which gives an overview of its execution.

5.2.1 Parameters

As the goal is to precisely locate the autonomous vehicle in a 2D environment, the state vector will now contain both position and velocity estimates for the x and y coordinates:

$$\hat{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix}$$

where x_k and y_k are the estimated position coordinates, and v_{x_k} and v_{y_k} are the estimated velocities in the x and y directions at time step k .

Assuming that the vehicle is moving with constant velocity, the state transition matrix is

given by:

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

This matrix accounts for the fact that the next position depends on both the current position and the velocity, where Δt is the time interval between steps.

The process noise covariance matrix Q_k represents how much uncertainty there is in the prediction of both position and velocity between time steps. Small values are assigned to reflect this uncertainty:

$$Q = \begin{bmatrix} q_x & 0 & 0 & 0 \\ 0 & q_y & 0 & 0 \\ 0 & 0 & q_{v_x} & 0 \\ 0 & 0 & 0 & q_{v_y} \end{bmatrix}$$

Here, q_x and q_y are small constants representing uncertainty in the position, and q_{v_x} and q_{v_y} represent uncertainty in the velocity.

DW1000 has accuracy specifications of ± 10 cm, which is used as an estimate for the standard deviation:

$$\sigma_x = \sigma_y = 0.1 \text{ meters}$$

So,

$$\sigma_x^2 = \sigma_y^2 = (0.1)^2 = 0.01$$

Therefore, the measurement noise covariance matrix R_k which accounts for the noise in the UWB measurements, is:

$$R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

The measurement matrix H_k relates the state vector (position and velocity) to the actual measurements, which only provide x and y positions:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

This indicates that the UWB measurements directly provide the x and y coordinates without velocity components.

The first measurement is set as the initial estimate of the position, and the initial velocity is assumed to be zero:

$$\hat{x}_0 = \begin{bmatrix} x_0 \\ y_0 \\ 0 \\ 0 \end{bmatrix}$$

5.2.2 Kalman Filter Algorithm

1: **function** KALMANFILTER(z_k)
 2: **Input:** z_k (current measurement of position)
 3: **Output:** \hat{x}_k (current state estimate including position and velocity)
 4: **Initialization:**
 5: Initialize the state estimate \hat{x}_0 and error covariance P_0
 6: **Prediction Step:**

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} && \text{(Predicting the state)} \\ P_k^- &= AP_{k-1}A^T + Q && \text{(Predicting the error covariance)}\end{aligned}$$

7: **Update Step:**
 8: **Step 1: Computing Kalman Gain**

$$K_k = P_k^- H^T \left(H P_k^- H^T + R \right)^{-1}$$

9: **Step 2: Updating the state estimate with the measurement**

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$$

10: **Step 3: Updating the estimate error covariance**

$$P_k = (I - K_k H)P_k^-$$

11: **Return** \hat{x}_k
 12: **end function**

5.3 Communication Protocol

A number of different protocols were employed to exchange message between different systems and subsystems such as SPI, I/O and TCP/IP. These protocols and their implementation have been discussed in the subsections below. Subsection 5.3.1 shows the pin-diagram highlighting the pins that were used for communication between ESP32 and the DW1000. Subsection 5.3.2, shows the connection between the Arduino Esp32 which is important in ensuring that the car knows when it has to turn while being in the zone. Finally, 5.3.3, discusses the importance of TCP/IP in ensuring remote visualisation of the coordinates of the car while it is moving.

5.3.1 SPI

The DW1000 UWB modules communicate with the ESP32 using the Serial Peripheral Interface (SPI) protocol. SPI is chosen for its high-speed data transfer capabilities, enabling efficient exchange of ranging data between the UWB modules and the ESP32. This ensures accurate distance measurements, which are critical for real-time positioning in this project. As shown in fig. 5.4, IO 18,19, 23 and 4 were used for Clock, MISO, MOSI, Chip Select respectively. The Arduino Library is used to initiate the SPI communication with SPI.begin().

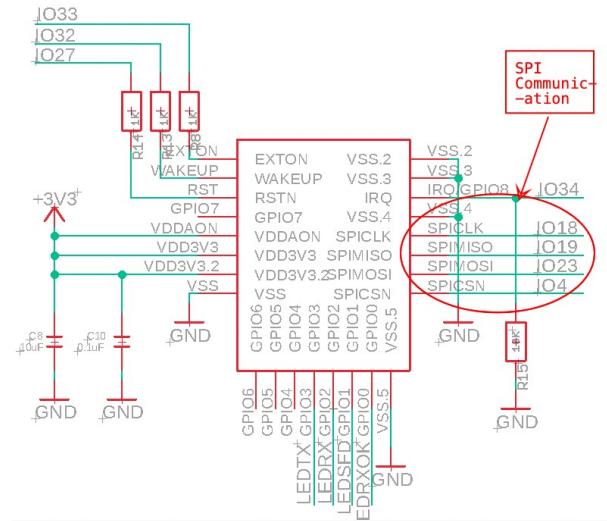


Figure 5.4: SPI Communication [Mak24]

5.3.2 I/O

IO 12 of ESP32 is connected to Analog Pin A5 of Arduino Uno. When the tag reaches the desired zone where $1.8 \leq x \leq 2.5$ and $3.4 \leq y$, IO 12 of ESP32 goes HIGH from LOW meaning the Analog Pin A5 goes HIGH as well. As A5 is high, the prototype now knows that it has reached the zone and should take a U-turn now.

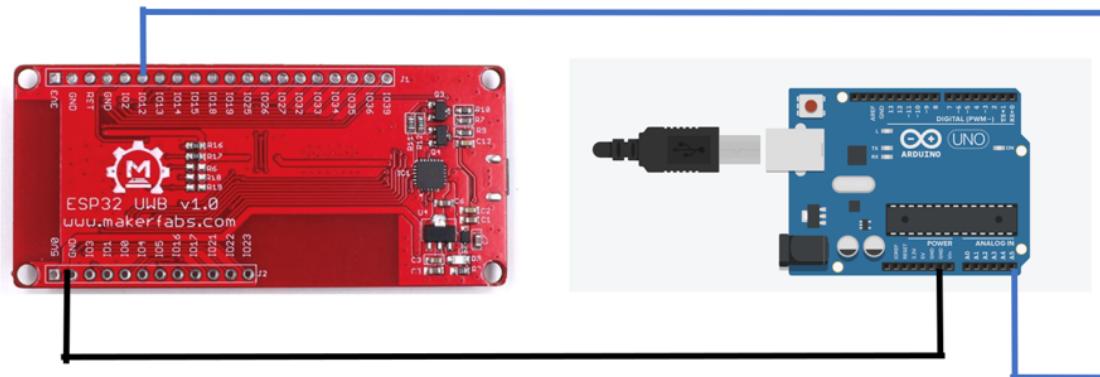


Figure 5.5: Pin Connection between Arduino and ESP32UWB

5.3.3 TCP/IP

In this implementation, TCP (Transmission Control Protocol) is utilized to establish a reliable connection between a server and a client for real-time data communication. The server is configured using Python's `socket` library, where `socket.AF_INET` specifies IPv4 addressing, and `socket.SOCK_STREAM` indicates the use of TCP's connection-oriented protocol. The server binds to a specified IP address (`0.0.0.0`, allowing it to listen on all network interfaces) and port (8080), and it enters a listening state using the `listen()`

method, awaiting incoming client connections. Once a connection is accepted, the server continuously receives data in fixed-size buffers via the `recv()` method. TCP ensures that data packets are delivered in sequence and without loss, enabling accurate and consistent transmission of the coordinates sent by the client. This reliable communication facilitates the integration of real-time data into a Kalman filter for smooth visualization and processing.

6 Evaluation

Based on what was read in chapter 3 and the analysis of chapter 4, the implementation of the project was discussed in chapter 5. The purpose of this chapter is to visualise and evaluate the results received in chapter 5. This chapter is divided into four portions or sections. Section 6.1 talks about the readings received with the car being stationary at one place. On the other hand section 6.2 presents the data with the car moving on the track. The shape of the track can be found in fig. 5.1. With the readings shown in section 6.1 & 6.2, further improvements were performed by tweaking the threshold value and the parameters of Kalman Filter. Finally, in section 6.4, the movement of the track is tracked on the car where it is expected to ignore the zone of $1.8 \leq x \leq 2.5$ and $3.4 \leq y$.

6.1 Stationary Tag

As the name already suggests, this section presents the readings when the car is stationary on the track. The purpose of this practical verification is to check the preciseness of the sensor or the noise level from the surroundings. This section is divided into two portions: subsection 6.1.1 discusses LOS scenario which can be seen in fig. 5.2 & subsection 6.1.2 discusses NLOS scenario which can be found in fig. 5.3.

6.1.1 LOS

As shown in figure 6.2, even with the tag remaining stationary in a line-of-sight (LOS) scenario, the received coordinates become significantly jittery after one minute due to environmental noise. The estimated position of the tag in this case is approximately $(1.7, 1.1) \pm 0.2$ meters, which falls short of expectations, considering the DW1000 is capable of precision up to 0.1 meters. To address this issue, a Kalman Filter was applied, and the improved results are displayed in Figure 6.2. With the Kalman Filter, R matrix is recorded as 0.01 and Q matrix recorded as 0.0000001. , the estimated position after one minute stabilizes to $(1.74, 1.24) \pm 0.01$ meters, demonstrating a significant reduction in noise and a marked improvement in precision.

6.1.2 NLOS

In a non-line-of-sight (NLOS) scenario, as shown in Figure 6.3, the readings after one minute exhibit increased noise, with the coordinates estimated at approximately $(1.7, 0.9) \pm 0.2$ meters. Given this, the Kalman Filter was applied again, and the results are presented in Figure 6.4. With the similar value in R and Q matrix recorded above, the filtered coordinates stabilize at $(1.65, 1.40) \pm 0.05$ meters, achieving the expected precision of within 10 cm.

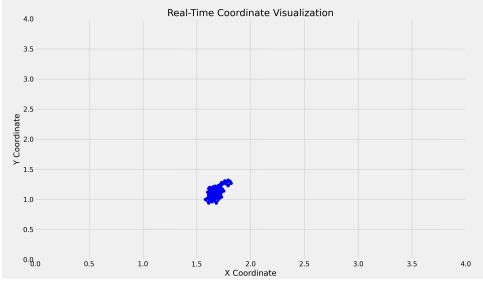


Figure 6.1: Stationary Tag in LOS without Kalman Filter

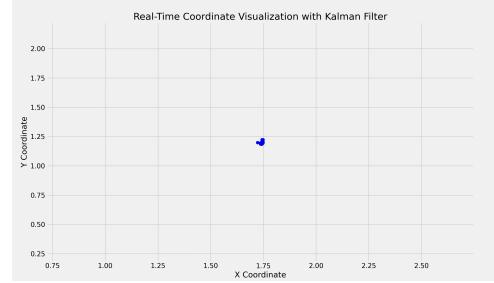


Figure 6.2: Stationary Tag in LOS with Kalman Filter

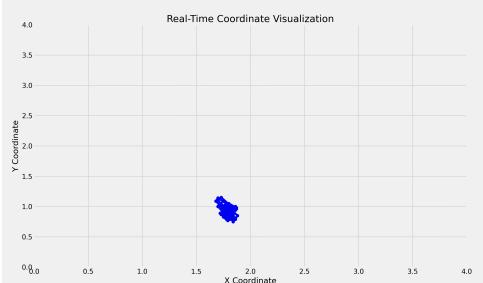


Figure 6.3: Stationary Tag in NLOS without Kalman Filter

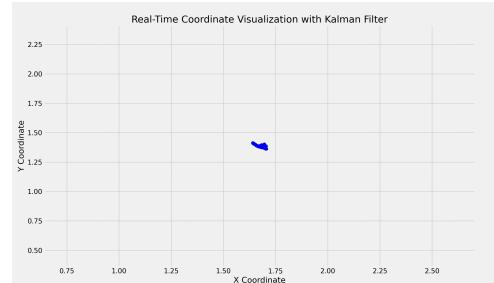


Figure 6.4: Stationary Tag in NLOS with Kalman Filter

6.2 Moving Tag

This section presents the readings when the car is moving on the track. The readings received allowed to do tweaking for the final part of the project. This section is divided into two portions: subsection 6.2.1 discusses LOS scenario which can be seen in fig. 5.2 & subsection 6.2.2 discusses NLOS scenario which can be found in fig. 5.3.

6.2.1 LOS

Figure 6.7 shows the movement of the autonomous vehicle with the tag mounted on top. The presence of noise is evident, with spikes in the data causing readings to deviate from the intended track. These deviations are expected due to process and measurement noise. The Kalman Filter is applied to mitigate this noise, ensuring more accurate tracking. Various values for the R and Q matrices were tested to optimize the results with the moving tag. Figure 6.6 displays the readings after optimization, with the R matrix value set to 0.1 and the Q matrix value set to 0.001. This configuration was found to be optimal, significantly reducing noise compared to the unfiltered data. As can be seen in 6.7, a huge spike is seen that goes way outside the range of the track. In order to completely eliminate these unrealistic readings, a function was created that ignores any readings outside the x coordinate of 0 to 4. The result can be seen in 6.8, as no unrealistic spikes are seen and values are inside the track. Therefore to reduce spikes and frequent noise, Kalman Filter was integrated a noise-cancelling algorithm that ensures that readings are always within the expected range. This implementation is shown in 6.3

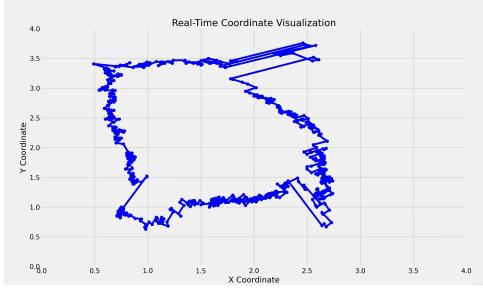


Figure 6.5: Moving Tag in LOS without Kalman Filter

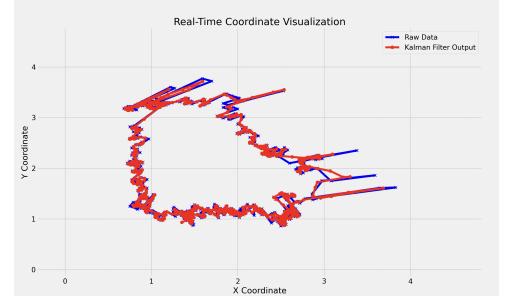


Figure 6.6: Moving Tag in LOS with Kalman Filter

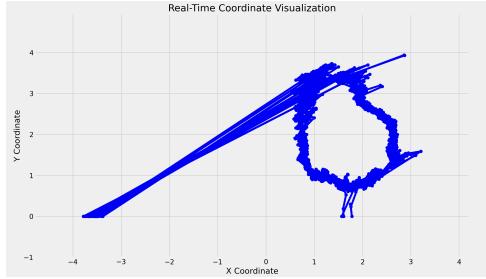


Figure 6.7: Moving Tag in LOS without threshold algorithm

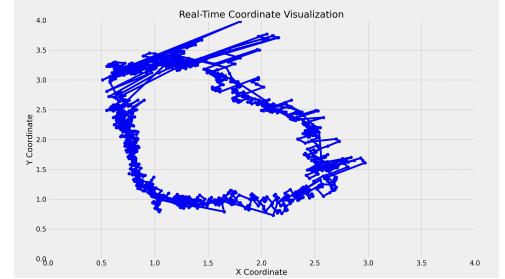


Figure 6.8: Moving Tag in LOS with constrained coordinates

6.2.2 NLOS

With NLOS, noise and imprecise readings were expected. As shown in fig. 6.9, readings go outside the track more frequently than ever before and the track shape is almost impossible to identify from this graph. With the integration of Kalman Filter, frequent noise spikes are reduced and an idea of the track's shape can be somewhat figured out.

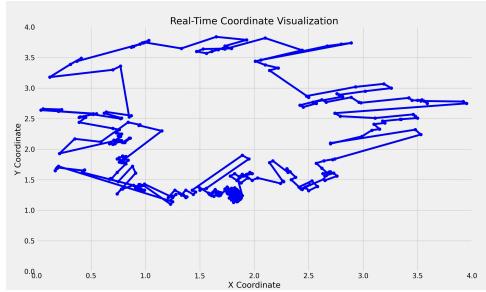


Figure 6.9: Moving Tag in NLOS without Kalman Filter

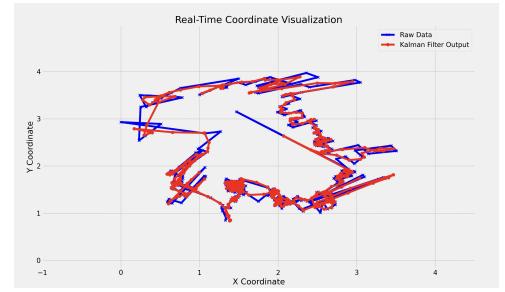


Figure 6.10: Moving Tag in NLOS with Kalman Filter

6.3 Improvisation

Finally, Kalman filter with the integration of threshold algorithm has been applied while the Tag is moving in the track. As can be observed in fig. 6.11, No significant spike can be seen and the track shape is much more clear. This implementation is used in the final part of project where the vehicle is expected to keep track of current coordinates at all

times. This is shown in fig. 6.13 where the vehicle will move around the track and make self-decisions based on the coordinate calculated by Anchor. The mixture of Kalman Filter and threshold-algorithm is expected to make the coordinate reading precise for which the vehicle can make accurate decisions.

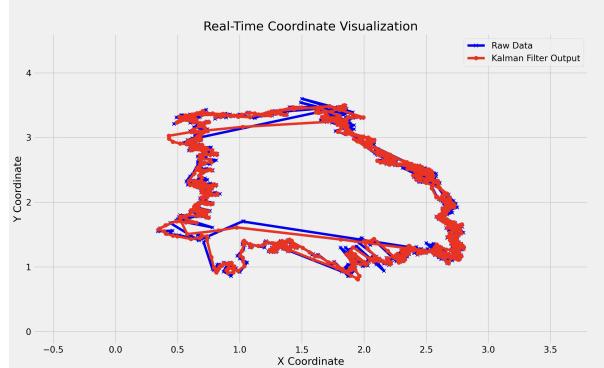


Figure 6.11: Moving Tag in LOS with Kalman Filter and Threshold Algorithm

6.4 Self-localization

As shown in fig. 6.13 & fig. 6.12, at $1.8 \leq x \leq 2.5$ and $3.4 \leq y$, the vehicle is taking a U-turn. The gap at the plot means that the vehicle has always ignored those points as it took a U-turn.

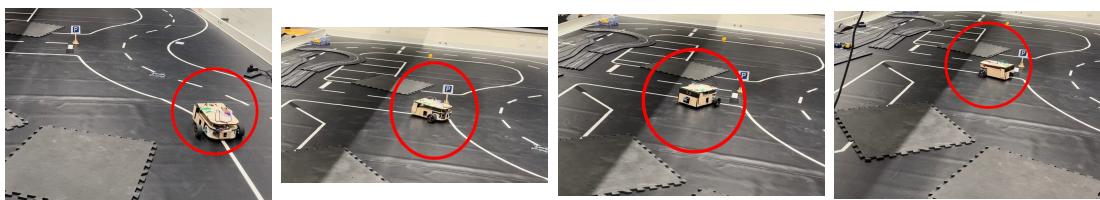


Figure 6.12: Prototype taking U-turn after reaching target.

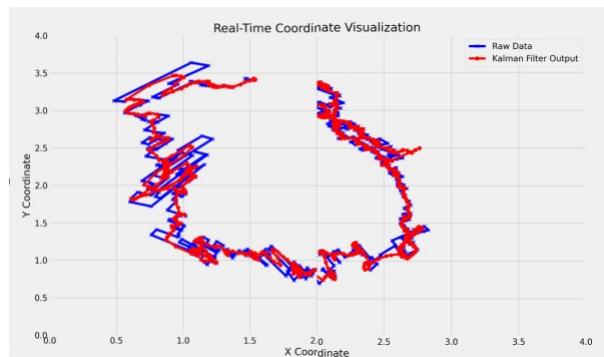


Figure 6.13: Car taking U-turn when reaching destination

7 Outlook and Summary

With the evaluation shown in chapter 6, it can be said that UWB technology is crucial for indoor localisation at cm level precision. The readings of LOS in both stationary and non-stationary scenarios showed that the sensor receives noise from the surroundings making it precise to 20 cm. The addition of Kalman Filter improves this ensuring that the precision is within 10 cm. Even with the precision caused by filter, some major spikes were seen when the vehicle was moving. These spikes were removed by a threshold algorithm which ensures that there is a realistic difference between the current and previous reading. This allowed to ensure that the vehicle takes turns in the expected zone of $1.8 \leq x \leq 2.5$ and $3.4 \leq y$. The vehicle made a turn everytime at the range of x coordinate mentioned above meaning that the sensor is accurate up to 7 cm.

For future work, I would like to find out real-time 3D coordinates including the z-axis. For that more anchors would be necessary and with that there will be more challenges to overcome. Therefore, other Kalman Filter algorithms such as EKF can be used to improve the results. Additionally, as this paper discussed primarily the monitoring of the vehicle, I would like to manipulate the movement of autonomous vehicle based on coordinate based instructions, ensuring precise remote control of the movement of the vehicle. Finally, another idea that I thought of is where another tag will be placed at a certain point which will eliminate the need to telling the vehicle to take U-turn at specifically $1.8 \leq x \leq 2.5$ and $3.4 \leq y$. The use of another tag as an indicator of a U-turn zone can allow us to change the zone dynamically in real-time.

8 Acknowledgements

The research and experiment of the topic has been possible due to the constant feedback and guidance of Prof. Dr. Henkler and Lukas Walter. Additional thanks to Hamm-Lippstadt University of Applied Sciences for providing the needed material and working environment (lab). Additional shoutout to Makerfabs for the DW1000 development board and learning materials provided online in <https://www.makerfabs.cc/article/esp32-uwb-indoor-positioning-test.html> and also to jremington for the DW1000 library and source code for tag and anchor here https://github.com/jremington/UWB-Indoor-Localization_Arduino

Bibliography

- [ARR⁺07] Michal Aftanas, Jana Rovnakova, Maria Riskova, Dusan Kocur, and Milos Drutarovsky. An analysis of 2d target positioning accuracy for m-sequence uwb radar system under ideal conditions. In *2007 17th International Conference Radioelektronika*, pages 1–6, 2007.
- [Bar00] Terence Barrett. History of ultrawideband (uwb) radar communications: Pioneers and innovators. 07 2000.
- [BCM⁺22] Tatiana A. Brovko, Alexander A. Chugunov, Alexander P. Malyshev, Ilya V. Korogodin, Oleg V. Glukhov, and Elena V. Silayeva. Analytical estimation of kalman filter errors for indoor positioning systems. In *2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, pages 1–5, 2022.
- [BKH18] Biki Barua, Nahi Kandil, and Nadir Hakem. On performance study of twr uwb ranging in underground mine. In *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pages 28–31, 2018.
- [BSY⁺23] Nuradlin Borhan, Izzati Saleh, Azan Yunus, Wan Rahiman, Dony Novaliendry, and Risfendra. Reducing uwb indoor localization error using the fusion of kalman filter with moving average filter. In *2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 55–59, 2023.
- [Bun16] Bundesnetzagentur. Vfg. 37/2016 allgemeinzuteilung von frequenzen für die nutzung durch ultrabreitband-anwendungen (uwb), 2016. Official Document.
- [CYH21] Junhui H. Cheng, Peng P. Yu, and You R. Huang. Application of improved kalman filter in under-ground positioning system of coal mine. *IEEE Transactions on Applied Superconductivity*, 31(8):1–4, 2021.
- [EU-07] EU-Kommission. Entscheidung der kommission vom 21. februar 2007 über die gestattung der harmonisierten funkfrequenznutzung für ultra-breitbandgeräte in der gemeinschaft, 2007. Official Document.
- [Fir24] Fira. The present and future of uwb. sep 2024.
- [Gri21] Eric Griffith. How do apple airtags work? ultra-wideband explained. sep 2021.

- [IEE11] IEEE Standards Association. IEEE Standard for Local and metropolitan area networks. Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANS), 2011. IEEE Std 802.15.4-2011.
- [IKS21] Youssef Ibnatta, Mohammed Khaldoun, and Mohammed Sadik. Indoor localization techniques based on uwb technology. In Halima Elbiaze, Essaid Sabir, Francisco Falcone, Mohamed Sadik, Samson Lasaulce, and Jalel Ben Othman, editors, *Ubiquitous Networking*, pages 3–15, Cham, 2021. Springer International Publishing.
- [KY10] Hakan Koyuncu and Shuang-Hua Yang. A survey of indoor positioning and object locating systems. *International Journal of Computer Science and Network Security (IJCSNS)*, 10, 01 2010.
- [LZB⁺21] Ying Liu, Xiaozhang Zhu, Xiaole Bo, Ziyan Yang, and Zhiqin Zhao. Ultra-wideband high precision positioning system based on sds-twr algorithm. In *2021 International Applied Computational Electromagnetics Society (ACES-China) Symposium*, pages 1–2, 2021.
- [Mak24] Makerfabs. Makerfabs esp32 uwb - hardware. <https://github.com/Makerfabs/Makerfabs-ESP32-UWB/tree/main/hardware>, 2024. Accessed: 2024-11-26.
- [MX21] Tien Le Minh and Dung Trinh Xuan. Applying kalman filter to uwb positioning with ds-twr method in los/nlos scenarios. In *2021 International Symposium on Electrical and Electronics Engineering (ISEE)*, pages 95–99, 2021.
- [RVdVSDP18] Matteo Ridolfi, Samuel Van de Velde, Heidi Steendam, and Eli De Poorter. Analysis of the scalability of uwb indoor localization solutions for high user densities. *Sensors*, 18(6), 2018.
- [SDM14] Stephan Sand, Armin Dammann, and Christian Mensing. *Positioning in Wireless Systems*, pages 189–192. Springer, 2014.
- [Sne19] Jason Snell. The u1 chip in the iphone 11 is the beginning of an ultra wideband revolution. sep 2019.
- [SXW⁺23] Boyang Sun, Yanfeng Xia, Xiaofeng Wang, You Zhang, Ruanming Huang, Shuhan Wang, and Aimin Tang. A reliable and efficient uwb-based indoor localization scheme. In *2023 IEEE 15th International Conference on Advanced Infocomm Technology (ICAIT)*, pages 186–191, 2023.
- [SYG19] Yu Songcan, Liu Yongxin, and Li Gen. Twice correction indoor positioning algorithm using ultra-wideband technology. In *2019 Chinese Automation Congress (CAC)*, pages 3748–3753, 2019.
- [TSB17] Simon Tewes, Ludwig Schwoerer, and Patrick Bosselmann. Designing a basic ir-uwb-rtls - raw-data position estimation utilizing twr. In *Smart SysTech 2017; European Conference on Smart Objects, Systems and Technologies*, pages 1–5, 2017.

- [VHVLVLDP24] Ben Van Herbruggen, Dries Van Leemput, Jaro Van Landschoot, and Eli De Poorter. Real-time anchor node selection for two-way-ranging (twr) ultra-wideband (uwb) indoor positioning systems. *IEEE Sensors Letters*, 8(3):1–4, 2024.
- [Wor19] RF Wireless World. Uwb vs gps | difference between uwb and gps. 2019.
- [WTMF⁺18] Jing Wang, Yao Tang, José-María Muñoz-Ferreras, Roberto Gómez-García, and Changzhi Li. An improved indoor localization solution using a hybrid uwb-doppler system with kalman filter. In *2018 IEEE Radio and Wireless Symposium (RWS)*, pages 181–183, 2018.

List of Figures

| | | |
|------|--|----|
| 1.1 | Prototype of Autonomous Vehicle | 2 |
| 1.2 | Car Following Line | 2 |
| 2.1 | UWB frequency spectrum [IKS21] | 5 |
| 2.2 | DS-TWR concept [SYG19] | 8 |
| 2.3 | Triangle with all known sides | 9 |
| 2.4 | Kalman Filter Recursion | 10 |
| 2.5 | Overall Flow Chart | 11 |
| 4.1 | ESP32UWB | 16 |
| 4.2 | Requirement Diagram | 18 |
| 4.3 | Use Case Diagram | 19 |
| 4.4 | Sequence Diagram | 19 |
| 4.5 | Block Definition Diagram | 20 |
| 4.6 | Activity Diagram | 20 |
| 5.1 | 2D Setup | 21 |
| 5.2 | LOS | 22 |
| 5.3 | NLOS | 22 |
| 5.4 | SPI Communication [Mak24] | 25 |
| 5.5 | Pin Connection between Arduino and ESP32UWB | 25 |
| 6.1 | Stationary Tag in LOS without Kalman Filter | 28 |
| 6.2 | Stationary Tag in LOS with Kalman Filter | 28 |
| 6.3 | Stationary Tag in NLOS without Kalman Filter | 28 |
| 6.4 | Stationary Tag in NLOS with Kalman Filter | 28 |
| 6.5 | Moving Tag in LOS without Kalman Filter | 29 |
| 6.6 | Moving Tag in LOS with Kalman Filter | 29 |
| 6.7 | Moving Tag in LOS without threshold algorithm | 29 |
| 6.8 | Moving Tag in LOS with constrained coordinates | 29 |
| 6.9 | Moving Tag in NLOS without Kalman Filter | 29 |
| 6.10 | Moving Tag in NLOS with Kalman Filter | 29 |
| 6.11 | Moving Tag in LOS with Kalman Filter and Threshold Algorithm | 30 |
| 6.12 | Prototype taking U-turn after reaching target. | 30 |
| 6.13 | Car taking U-turn when reaching destination | 30 |

List of Tables

| | | |
|-----|---|----|
| 4.1 | Overview of frequency parameters according to [Bun16] | 15 |
| 4.2 | Ranging Offset | 17 |

Affidavit

I, Rubayet Kamal, herewith declare that I have composed the present paper and work by myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance. .

Lippstadt, December 11, 2024

Rubayet Kamal

R.K.Erhardt