

UWB BASED DRONE LOCALIZATION FOR AUTONOMOUS LANDING
SYSTEMS

by

Şener Yılmaz

B.S., Electrical & Electronics Engineering, Bilecik Şeyh Edebali University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical & Electronics Engineering
Boğaziçi University
2025

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Professor Ahmet Öncü for his invaluable insights and unwavering support throughout the course of this research. His guidance and expertise have been instrumental in shaping the direction and outcome of this thesis.

I am profoundly thankful to my family for their unconditional support, encouragement, and belief in me throughout this journey. Their love and understanding have been a source of strength and motivation.

I am also deeply grateful to the faculty members of the Electrical and Electronics Engineering Department and the Computer Engineering Department at Boğaziçi University. The knowledge and skills I have gained during my time at the university have been invaluable in shaping both my academic and professional growth.

Finally, I extend my heartfelt thanks to my coworkers and friends for their continuous support, collaboration, and fellowship. Their encouragement and shared enthusiasm have made this journey not only productive but also memorable.

ABSTRACT

UWB BASED DRONE LOCALIZATION FOR AUTONOMOUS LANDING SYSTEMS

Accurate localization is essential for autonomous unmanned aerial vehicle (UAV) operations, particularly during critical maneuvers such as landing. This thesis presents a robust, high-accuracy localization system utilizing Ultra-Wideband (UWB) technology combined with sensor fusion. The proposed system employs UWB anchors arranged compactly on a $0.6\text{m} \times 0.7\text{m}$ platform and a customized UWB tag equipped with inertial orientation sensors for precise positioning. Real-time data processing, including sensor calibration, signal filtering, and fusion, is implemented on a Raspberry Pi 5 microcomputer.

Orientation estimation is performed using the Versatile Quaternion-based Filter (VQF) algorithm, which integrates calibrated data from accelerometer, gyroscope, and magnetometer sensors. To enhance distance measurement accuracy, a multilayer perceptron (MLP) model is trained and applied, significantly reducing the UWB ranging error from 9.26 cm to 4.35 cm root mean square error (RMSE). An Extended Kalman Filter (EKF) further integrates these corrected UWB measurements with dynamic acceleration and orientation data to achieve resilient and accurate localization.

Experimental evaluations confirm the system's high performance, demonstrating an average RMSE of 4.28 cm in static scenarios. Dynamic tests validate reliable tracking capabilities across diverse trajectories, underscoring the system's suitability for autonomous UAV landing applications. The presented research provides a solid groundwork for future developments in sensor calibration, data integration, and advanced localization algorithms.

ÖZET

OTONOM İNİŞ SİSTEMLERİ İÇİN UWB TABANLI DRON KONUMLANDIRMA

Doğru konumlandırma, özellikle iniş gibi kritik manevralarda, otonom insansız hava araçları (İHA) operasyonlarının güvenliği ve başarısı için kritik öneme sahiptir. Bu tez çalışmasında, Ultra Geniş Bant (UWB) teknolojisi ile sensör füzyonunu birleştiren dayanıklı ve yüksek doğruluklu bir konumlandırma sistemi geliştirilmiştir. Önerilen sistem, $0.6\text{m} \times 0.7\text{m}$ boyutlarındaki kompakt bir platform üzerine yerleştirilmiş UWB referans noktaları ve yönelim sensörleriyle donatılmış özel bir UWB etiketi kullanarak hassas konumlandırma sağlamaktadır. Gerçek zamanlı sensör kalibrasyonu, sinyal filtreleme ve veri füzyonu işlemleri Raspberry Pi 5 mikro bilgisayar üzerinde gerçekleştirilmiştir.

Yönelim tahmini, kalibre edilmiş ivmeölçer, jiroskop ve manyetometre verilerinin birleştirildiği VQF algoritması ile yapılmıştır. Ölçüm doğruluğunu artırmak amacıyla eğitilen çok katmanlı algılayıcı (MLP) modeli sayesinde, UWB mesafe ölçümündeki hata oranı 9.26 cm RMSE'den 4.35 cm'ye düşürülmüştür. Genişletilmiş Kalman Filtresi (EKF) ile düzeltilmiş UWB ölçümleri, dinamik ivme ve yönelim bilgileriyle entegre edilerek, dirençli ve yüksek hassasiyetli bir konumlandırma elde edilmiştir.

Yapılan deneysel değerlendirmeler, sistemin statik koşullarda ortalama 4.28 cm'lik RMSE ile yüksek doğrulukta çalıştığını göstermektedir. Dinamik testler ise farklı yörüngelerde güvenilir izleme yeteneğini doğrulayarak, sistemin otonom İHA iniş görevleri için uygunluğunu ortaya koymuştur. Bu araştırma, sensör kalibrasyonu, veri entegrasyonu ve gelişmiş konumlandırma algoritmaları alanındaki gelecekteki çalışmalar için sağlam bir temel sunmaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF SYMBOLS	xiii
LIST OF ACRONYMS/ABBREVIATIONS	xiv
1. INTRODUCTION	1
1.1. Research Background	1
1.2. Problem Statement	4
1.3. Research Objectives	5
1.4. Previous Work	6
1.5. Scope	7
1.6. Contributions	8
1.7. Thesis Structure	9
2. BACKGROUND	11
2.1. UWB Technology	11
2.1.1. Signal Models in Ultra-Wideband (UWB) Systems	12
2.1.1.1. Carrier-Based UWB (OFDM-UWB)	14
2.1.1.2. Time-Hopping Spread Spectrum (TH-UWB)	14
2.1.2. Pulse Modulation and Coding in Ultra-Wideband (UWB) Systems	15
2.1.2.1. Coding Techniques in UWB	16
2.1.3. UWB Regulations	17
2.1.4. IEEE Standards for IR-UWB	19
2.2. UWB Market Trends	20
2.3. Ranging techniques	21
2.3.1. Two way ranging (TWR)	21
2.3.2. Time difference of arrival (TDOA)	23

2.3.3.	Angle of Arrival (AoA)	24
2.3.4.	Received Signal Strength (RSS)	26
2.4.	Machine Learning (ML) methods	27
2.4.1.	Linear Regression	27
2.4.2.	Polynomial Regression	28
2.4.3.	Multilayer Perceptron (MLP)	29
2.4.4.	Alternative Machine Learning Methods	30
2.5.	Localization Algorithms	31
2.5.1.	Multilateration	31
2.5.2.	Kalman Filter	33
2.5.3.	Extended Kalman Filter (EKF)	35
2.5.4.	Alternative Localization Algorithms	37
3.	METHODOLOGY	38
3.1.	System Overview	38
3.2.	Hardware Components	40
3.2.1.	UWB transceiver and microcontroller	40
3.2.2.	Inertial Measurement Unit (IMU)	43
3.2.3.	Laser Sensor	44
3.2.4.	Raspberry Pi 5 Single Board Computer	45
3.3.	Calibration	47
3.3.1.	Inertial Measurement Unit (IMU) Calibration	47
3.3.2.	UWB Calibration	51
3.4.	Real-time Implementation	60
3.4.1.	Initialization and Configuration of the UWB Transceiver	60
3.4.2.	Sending and Receiving UWB Messages	62
3.4.3.	Modified Two-Way Ranging Protocol	63
3.4.4.	Data Flow and Communication	66
3.4.5.	Location Estimation	68
3.4.5.1.	Dynamic Acceleration Filtering	69
3.4.5.2.	Acceleration Vector Rotation	69
3.4.5.3.	Extended Kalman Filter Matrices	70

4. SYSTEM EVALUATION	73
4.1. Experimental Setup	73
4.2. Indoor Static Localization Test Results	74
4.3. Indoor Dynamic Localization Test Results	76
4.4. Outdoor Field Test Results	77
5. CONCLUSION	84
REFERENCES	86



LIST OF FIGURES

Figure 1.1.	UAV market share by industry.	2
Figure 2.1.	Common pulse shapes in ultra-wideband (UWB) systems.	13
Figure 2.2.	UWB -41.3 dBm/MHz limit compared to narrowband standards. . .	18
Figure 2.3.	Global UWB chipset market forecast (2022-2030).	20
Figure 2.4.	DS-TWR process.	21
Figure 2.5.	AoA estimation.	24
Figure 3.1.	Anchor coordinates and positioning coordinate system.	38
Figure 3.2.	DWM1001-DEV from Qorvo.	40
Figure 3.3.	DWM1001 module internal structure.	41
Figure 3.4.	UWB tag hardware.	42
Figure 3.5.	GY-85 sensor board.	43
Figure 3.6.	Benewake TFmini Plus laser sensor.	45
Figure 3.7.	Raspberry Pi 5 single board computer.	46
Figure 3.8.	Hard iron and soft iron effects on magnetometer readings.	47

Figure 3.9.	Magneto calibration software.	48
Figure 3.10.	Contents of magnetometer reading file as example.	50
Figure 3.11.	Orientation estimation using the calibrated IMU.	51
Figure 3.12.	Normalized CIR magnitudes in a multipath scenario.	53
Figure 3.13.	RSL estimation compared to actual RSL using DW1000 transceiver.	54
Figure 3.14.	MLP training setup with the tag and two anchors.	57
Figure 3.15.	Laser and UWB distance measurements during data collection.	57
Figure 3.16.	Close-up view of laser and UWB distance measurements.	58
Figure 3.17.	MLP layers.	59
Figure 3.18.	Close-up view of laser and UWB distance measurements after MLP correction.	59
Figure 3.19.	DW1000 initialization flow diagram.	61
Figure 3.20.	TWR process with multiple anchors.	63
Figure 3.21.	Tag TWR algorithm.	64
Figure 3.22.	Anchor TWR algorithm.	65
Figure 3.23.	Data acquisition system diagram.	67

Figure 3.24.	Locationing flowchart.	68
Figure 4.1.	Experimental setup featuring UWB anchors.	73
Figure 4.2.	Static localization test with UWB tag placed at $(0, 0)$	74
Figure 4.3.	Static localization test with UWB tag placed at $(-0.66, -0.66)$	74
Figure 4.4.	All measured and expected locations during static localization test.	75
Figure 4.5.	Measured trajectory of the UWB tag during the square path dynamic localization test.	76
Figure 4.6.	Measured trajectory of the UWB tag during the simulated landing trajectory test.	77
Figure 4.7.	DJI Matrice 300 RTK drone employed in field tests.	78
Figure 4.8.	Configuration of the localization system during outdoor field tests.	79
Figure 4.9.	UWB tag manually placed at the origin $(0 \text{ m}, 0 \text{ m})$	80
Figure 4.10.	UWB tag manually placed at coordinates $(5 \text{ m}, 0 \text{ m})$	81
Figure 4.11.	UWB tag manually placed at coordinates $(7.5 \text{ m}, 0 \text{ m})$	81
Figure 4.12.	Drone during mid-flight maneuvers.	82
Figure 4.13.	Drone at the landing platform center.	82

LIST OF TABLES

Table 3.1.	UWB configuration parameters used in this work.	61
Table 4.1.	Static measurement results	75



LIST OF SYMBOLS

a	Acceleration in the body or world frame (m/s ²)
A^{-1}	Soft-iron calibration matrix for magnetometer
c	Speed of light in the medium (m/s)
d	Distance between tag and anchor (m)
F	State transition Jacobian matrix in the EKF
g	Gravitational acceleration (9.81 m/s ²)
h	Raw magnetometer measurement vector
h_{cal}	Calibrated magnetometer measurement vector
H	Observation (measurement) Jacobian matrix in the EKF
b	Hard-iron bias vector for magnetometer calibration
P	State (estimation) covariance matrix
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
t	Time (s)
v	Speed or velocity magnitude (m/s)
x, y	Position coordinates in the ENU (or chosen) frame (m)
q	Quaternion representing orientation in 3D space
α	Blending parameter for high-pass filtering
Δt	Time step between successive filter updates (s)
ϕ, θ, ψ	Roll, pitch, and yaw angles (rad) in UAV orientation
τ	Time constant for high-pass or low-pass filter (s)
ϑ	Angle of Arrival (AoA) (rad)
Θ	Parameter set in machine-learning model (weights, biases, etc.)

LIST OF ACRONYMS/ABBREVIATIONS

AoA	Angle of Arrival
BLE	Bluetooth Low Energy
CIR	Channel Impulse Response
DS-TWR	Double-Sided Two-Way Ranging
EKF	Extended Kalman Filter
ENU	East-North-Up
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
LOS	Line of Sight
MLP	Multi-Layer Perceptron
NLOS	Non-Line of Sight
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
TDOA	Time Difference of Arrival
ToF	Time of Flight
TWR	Two-Way Ranging
UAV	Unmanned Aerial Vehicle
UWB	Ultra-Wideband
VQF	Versatile Quaternion-based Filter

1. INTRODUCTION

This chapter sets the stage for the study by outlining the overall research context, identifying the problem area, and explaining the motivation behind the investigation. An overview of the main issues that are addressed in the subsequent sections is provided.

1.1. Research Background

Unmanned aerial vehicles (UAVs) have found widespread applications across various industries, revolutionizing traditional operations with their versatility and efficiency. In infrastructure inspection, UAVs provide safe and cost-effective solutions for monitoring power lines, bridges, and pipelines in hard-to-reach areas [1]. In agriculture, they enable precision farming by assessing crop health, mapping fields, and optimizing irrigation and pesticide application [2]. Logistics industries benefit from UAVs for last-mile delivery, particularly in urban or remote locations, streamlining supply chains [3]. In disaster management, UAVs are invaluable for search and rescue missions, damage assessment, and delivering critical supplies to inaccessible regions [4]. Additionally, UAVs are extensively used in aerial surveys, capturing high-resolution imagery and topographic data for land surveying, urban planning, and environmental monitoring.

Among the various challenges faced by UAV operations, the landing maneuver is one of the most critical. Autonomous landing, in particular, requires precise localization to ensure that UAVs can safely return to designated areas such as charging stations or operational bases. Failures during the landing phase can lead to severe consequences, including damage to the UAV, disruption of operations, and potential harm to nearby personnel and infrastructure. As a result, accurate and reliable localization systems are indispensable for mitigating these risks and ensuring safe landings.

Localization systems have been developed to address the challenges of precise

UAV positioning. Among these, the Global Navigation Satellite System (GNSS) and its enhanced variant, Real-Time Kinematic (RTK) GNSS, are commonly used. While GNSS provides meter-level accuracy, RTK GNSS can achieve centimeter-level precision, making it suitable for many UAV applications. However, both systems rely on satellite signals, which can be obstructed in indoor or urban environments [5]. This limitation makes them unreliable in areas such as warehouses, urban canyons, or under dense vegetation.

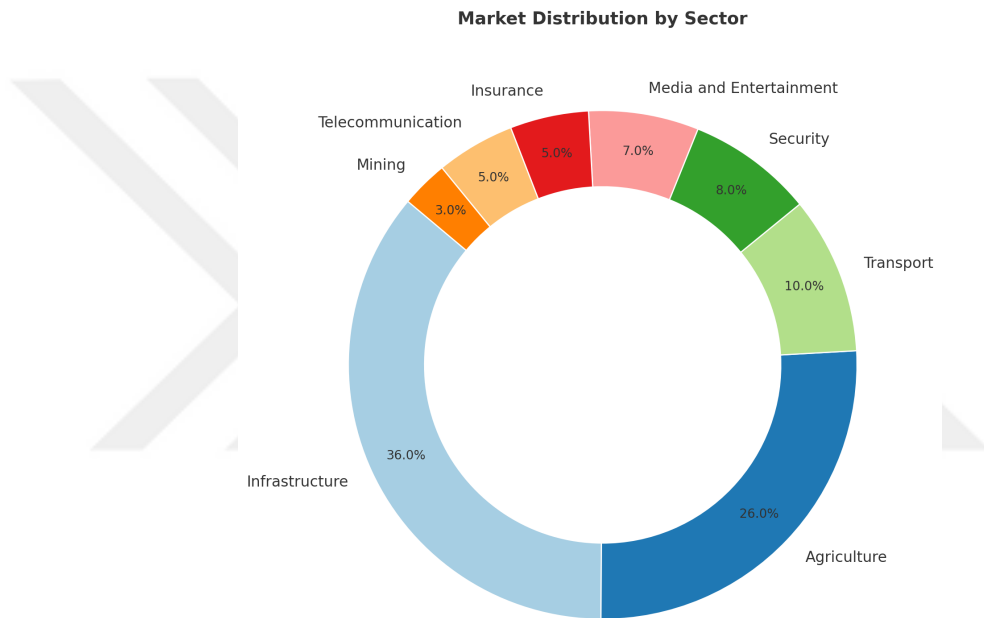


Figure 1.1. UAV market share by industry.

Vision-based systems offer another approach by utilizing cameras and image-processing algorithms to determine UAV position relative to markers or environmental features. While these systems are effective in well-lit and structured environments, they fail in low-light conditions, such as during nighttime operations, or in environments with smoke, fog, or dust. Additionally, vision-based systems require significant computational resources, increasing their complexity and limiting their suitability for real-time applications [6].

Millimeter-wave (mmWave) radar systems have also been explored for UAV lo-

calization due to their ability to detect objects in adverse environmental conditions. However, mmWave systems involve complex hardware setups and sophisticated software algorithms, making them challenging and costly to implement for many UAV applications [7].

Ultrasonic sensors, which measure distance by analyzing the time of flight or phase shift of sound waves, have been widely used in localization systems due to their cost-effectiveness, simplicity, and immunity to electromagnetic interference [8]. However, they exhibit several limitations, including a short measurement range (typically less than 10 meters) and susceptibility to environmental noise, such as wind and ambient vibrations. Furthermore, ultrasonic sensors are affected by variations in temperature and humidity, which can degrade accuracy if not compensated. While effective for basic non-contact distance measurements, these drawbacks restrict their applicability in high-precision UAV localization, particularly in dynamic or cluttered environments.

Ultra-Wideband (UWB) technology has emerged as a promising solution to overcome these limitations. UWB operates using short-duration radio pulses over a wide frequency spectrum, enabling it to provide accurate localization even in environments with significant electromagnetic interference or obstructions. Unlike GNSS, UWB does not rely on satellite signals, and unlike vision-based systems, it is unaffected by lighting conditions. Additionally, UWB systems can achieve high update rates and centimeter-level accuracy, making them ideal for real-time UAV localization and landing applications [9–13].

By addressing the weaknesses of other localization methods, UWB has proven to be a reliable and versatile technology for ensuring the safety and efficiency of UAV operations, particularly in critical tasks such as autonomous landing. This study builds upon these advantages, proposing an advanced UWB-based system that integrates high-accuracy positioning with robust sensor fusion and data processing techniques.

1.2. Problem Statement

The landing maneuver is one of the most critical phases in UAV operations, requiring precise and reliable localization to ensure safety and mission success. Even small errors during landing can result in crashes, damage to the UAV, or operational failure. This need for precision is amplified in scenarios where UAVs are tasked with landing on confined or mobile platforms, such as charging stations or moving vehicles. Therefore, developing a robust and accurate localization system is essential for enabling reliable autonomous landings.

While existing localization methods such as GNSS, vision-based systems, and mmWave radar provide partial solutions, they fall short in specific scenarios:

- **GNSS and RTK GNSS:** These systems are unreliable in GNSS-denied environments such as indoor facilities, urban canyons, or under dense vegetation.
- **Vision-Based Systems:** These systems depend heavily on favorable lighting conditions and clear visibility, making them unsuitable for low-light or obstructed environments.
- **mmWave Radar:** Despite their robustness in adverse conditions, these systems involve complex hardware and software setups, increasing cost and implementation challenges.
- **Ultrasonic Sensors:** Limited range and susceptibility to environmental noise restrict their use for high-precision UAV localization tasks.

UWB technology has emerged as a promising solution to address these challenges. With its ability to deliver high-accuracy localization even in environments with significant interference or non-line-of-sight conditions, UWB is well-suited for UAV applications. However, despite its advantages, UWB-based systems still face challenges such as:

- **Power consumption and device size constraints:** UWB transceivers, while

becoming more energy-efficient, still consume non-trivial amounts of power. For compact UAVs with limited battery life and strict payload weight budgets, integrating UWB modules without significantly impacting flight time or agility remains a hurdle.

- **Multipath propagation and environmental variability:** Although UWB is more resilient to multipath effects compared to other positioning technologies, rapidly changing environments—such as urban canyons, forests, or indoor spaces with moving obstacles—introduce signal distortions and propagation delays that complicate accurate localization.
- **Data fusion and algorithmic complexity:** To achieve the highest accuracy, UWB systems are often combined with other sensors (e.g., IMU (Inertial Measurement Unit), GPS, vision-based odometry) for sensor fusion. The complexity of designing and implementing robust fusion algorithms that can handle diverse conditions and gracefully degrade in challenging scenarios presents ongoing technical difficulties.

This thesis aims to address these gaps by developing a high-precision UWB-based localization system tailored for UAV autonomous landing.

1.3. Research Objectives

The primary objective of this research is to develop a robust and accurate localization system for UAV autonomous landing by leveraging UWB technology and advanced sensor fusion techniques. This system aims to address the limitations of existing localization methods, ensuring reliable performance in dynamic environments and under challenging conditions. To achieve this, the following objectives are defined:

- **Accurate Orientation Estimation:** Through precise calibration of IMU components (magnetometer, accelerometer, and gyroscope), the VQF (Versatile Quaternion-based Filter) orientation estimation algorithm is employed to fuse these sensor inputs, resulting in reliable device orientation estimates [14].

- **Machine Learning for Measurement Correction:** A pretrained Multi-Layer Perceptron (MLP) model is utilized to correct UWB measurement errors and outlier effects. The corrected measurements are then used to train a linear regression model with a sliding window for each iteration, reducing the impact of outliers and predicting measurements in case of failures.
- **Sensor Fusion with Extended Kalman Filter:** A tightly integrated Extended Kalman Filter (EKF) is employed to combine corrected UWB data with IMU measurements, enabling real-time and dynamic localization.
- **Achieving Real-Time Performance:** All data acquisition from anchors, machine learning (ML) corrections, and sensor fusion will be performed in real time, with the calculated location reported back to the UAV in near real time.
- **Real-World Validation:** The proposed system will be tested in real-world scenarios to evaluate its performance in terms of accuracy, reliability, and robustness under both static and dynamic conditions.

1.4. Previous Work

Numerous studies have explored UAV localization systems using UWB technology and IMU integration, providing valuable insights into the field.

Zeng et al. [15] proposed a UAV localization system tailored for landing on mobile platforms. Their setup consisted of four UWB anchors arranged on a 0.8×1.4 m landing platform and two UWB tags mounted on the UAV. A linear least squares (LLS) method was used for initial position estimation, followed by gradient descent optimization to refine the results. By fusing the position data with IMU measurements, their approach achieved a 61% reduction in root mean square error (RMSE) compared to standalone LLS localization.

Li et al. [16] introduced an EKF-based fusion of UWB and IMU data for 3D localization of MAV swarms in GNSS-denied environments. Their system deployed six UWB anchors in a cuboid configuration within an indoor exhibition hall. Each MAV

utilized time-of-arrival (TOA) measurements for localization, achieving 80 Hz real-time position updates. The system demonstrated high accuracy and scalability, suitable for dynamic swarm applications.

Strohmeier et al. [17] presented a UWB-based pose estimation system for small UAVs weighing less than 200 g. Their approach employed four UWB anchors in a minimal configuration to estimate 3D positions using double-sided two-way ranging (DS-TWR). By integrating UWB measurements with IMU and barometer data through a loosely coupled EKF, the system achieved an RMSE of 0.20 cm in position and 1.93° in orientation.

Ochoa-de-Eribe-Landaberea et al. [18] proposed a landing assistance system combining UWB, IMU, and magnetometer data for UAV localization on a 2×2 m platform. The system used two tags on the UAV and anchors deployed around the platform. A novel positioning algorithm fused tag data, reducing errors by 50% compared to traditional UWB systems. This approach demonstrated high robustness, even under sudden velocity changes during landing.

1.5. Scope

This research focuses on the development of a high-precision localization system for UAV autonomous landing, utilizing UWB technology and advanced sensor fusion methods. The system is designed specifically for scenarios requiring reliable performance in GNSS-denied environments or under dynamic operating conditions. The primary scope includes:

- **Technical Focus:**

- Integration of UWB anchors and tags for position estimation with high accuracy and low latency.
- Advanced calibration of inertial measurement unit (IMU) sensors, including magnetometer, accelerometer, and gyroscope.

- Implementation of machine learning techniques for correcting UWB diagnostics and mitigating outlier effects.
- Real-time sensor fusion using an EKF to combine UWB data and IMU measurements.
- **Operational Environment:**
 - The system is tailored for confined or mobile landing platforms, such as charging stations, and environments where GNSS or vision-based localization systems are impractical.
- **Exclusions:**
 - This work does not address vision-based localization, mmWave radar integration, or large-scale outdoor localization reliant on GNSS.

1.6. Contributions

This study introduces several advancements over prior research in UAV localization systems, outlined as follows:

- **Compact Anchor Deployment:** Unlike previous studies that often utilize larger platforms, this work achieves precise localization within a significantly smaller area, using UWB anchors mounted on a 0.6×0.7 m base.
- **Efficient System Architecture:** The localization system employs a single UWB tag on the UAV, with all computational processing offloaded to a microcomputer located on the base. This design reduces the onboard computational burden of the UAV, optimizing its operational efficiency.
- **Near Real-Time Localization and Orientation Estimation:** Through seamless communication between the UWB anchors and the UWB tag, the microcomputer provides near real-time 3D position and orientation information to the UAV. This enables precise and responsive localization, critical for autonomous landing maneuvers.
- **High-Accuracy Positioning with Machine Learning:** By leveraging machine learning techniques, the system uses a Multi-Layer Perceptron (MLP) model

to correct UWB measurements and mitigate outlier effects. Additionally, a linear regression model with a sliding window is employed to reduce the impact of outliers and predict measurements in cases of missing data, ensuring robust and accurate positioning.

By addressing gaps in existing solutions and providing a robust localization system, this research contributes to the safe and reliable operation of UAVs in autonomous landing scenarios.

1.7. Thesis Structure

This thesis is organized into five chapters, each of which addresses a specific aspect of the research and its outcomes.

Chapter 1 introduces the research topic, providing an overview of the background, problem statement, research objectives, scope, and contributions. It establishes the context and significance of the study while outlining the structure of the thesis.

Chapter 2 begins by providing an overview of UWB technology, detailing its fundamental principles and advantages in localization systems. The chapter then delves into distance measurement methods, explaining the underlying techniques and how they operate. Following this, machine learning methods relevant to localization are introduced, highlighting their role in enhancing measurement accuracy and robustness. Techniques used to compute positions from measured distances are also discussed, providing insights into their mathematical and practical implementations. Finally, the hardware components utilized in the proposed system are reviewed, emphasizing their functionality and integration within the localization framework.

Chapter 3 presents the methodology, detailing the design and implementation of the proposed localization system. This includes the calibration of IMU sensors, the integration of a machine learning model for measurement correction, and the develop-

ment of a sensor fusion framework using an EKF. The chapter also explains the system architecture and data processing steps.

Chapter 4 focuses on the evaluation of the proposed system. It describes the real-world testing setup and analyzes the system's performance in terms of accuracy, robustness, and reliability. The results are presented and discussed, highlighting the system's strengths and limitations.

Finally, Chapter 5 concludes the thesis by summarizing the research findings and discussing its contributions to the field of UAV localization. It also identifies limitations of the current work and proposes potential directions for future research.

2. BACKGROUND

The Background chapter provides the foundational knowledge required to understand the proposed UAV localization system. It begins with an overview of UWB technology, exploring its principles and advantages for high-precision localization. The chapter then discusses various ranging techniques used in localization, including Two-Way Ranging (TWR), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), and Received Signal Strength (RSS).

Next, the chapter introduces machine learning methods relevant to this research, such as Linear Regression, Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forests, and Multilayer Perceptrons (MLP). These methods are instrumental in improving measurement accuracy and reducing errors.

Finally, the chapter explores key localization algorithms, ranging from basic techniques like Trilateration and Multilateration to advanced filtering methods such as the Kalman Filter, Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter, providing a comprehensive understanding of their role in localization systems.

2.1. UWB Technology

The term "Ultra-Wideband" was officially adopted in the 1990s, as regulatory bodies and researchers sought to distinguish it from traditional narrowband and wideband systems. The Federal Communications Commission (FCC) played a pivotal role in the development of UWB technology by legalizing its commercial use in 2002, allocating the 3.1–10.6 GHz frequency band under strict power spectral density (PSD) constraints [19]. This decision spurred global research efforts, leading to the standardization of UWB-based communication and localization protocols.

Ultra-Wideband (UWB) technology is formally defined based on its spectral characteristics. According to the FCC's 2002 ruling, a signal is classified as UWB if it satisfies at least one of the following conditions [19]:

$$B > 500 \text{ MHz} \quad \text{or} \quad \frac{B}{f_c} > 0.2, \quad (2.1)$$

where B represents the absolute bandwidth, defined as the difference between the upper (f_H) and lower (f_L) frequencies at the -10 dB power level, calculated as $B = f_H - f_L$. The center frequency f_c is the arithmetic mean of these frequencies, given by

$$f_c = \frac{f_H + f_L}{2}. \quad (2.2)$$

2.1.1. Signal Models in Ultra-Wideband (UWB) Systems

Ultra-Wideband (UWB) communication relies on distinct signal models to efficiently utilize its large bandwidth while maintaining low power spectral density. Unlike traditional narrowband systems that use continuous sinusoidal carriers, UWB primarily employs impulse-based or spread-spectrum transmission schemes. These signal models enable UWB to achieve high temporal resolution, robustness to multipath effects, and efficient spectrum utilization.

The most common UWB signal model is Impulse Radio Ultra-Wideband (IR-UWB), which transmits short-duration pulses (on the order of nanoseconds or sub-nanoseconds) instead of modulated sinusoidal carriers. The transmitted signal can be expressed as

$$s(t) = \sum_{k=0}^{N-1} a_k p(t - kT_f), \quad (2.3)$$

where a_k is the amplitude of the k -th transmitted pulse, $p(t)$ represents the UWB pulse shape, T_f is the pulse repetition interval (PRI), and N denotes the total number of

transmitted pulses.

Common pulse shapes in Ultra-Wideband (UWB) systems include the Gaussian Monocycle, which is the first derivative of a Gaussian function and is given by

$$p(t) = -Ate^{-t^2/(2\sigma^2)}, \quad (2.4)$$

where A is the amplitude and σ controls the pulse width. Another widely used pulse shape is the Modified Hermite Pulse, which is often designed to meet spectral compliance requirements. Additionally, the Root Raised Cosine (RRC) Pulse is commonly employed for controlled spectral occupancy and efficient spectrum shaping.

The spectrum of an Impulse Radio Ultra-Wideband (IR-UWB) signal is extremely wide, ensuring fine time resolution and allowing accurate time-of-arrival (TOA) estimation in localization applications.

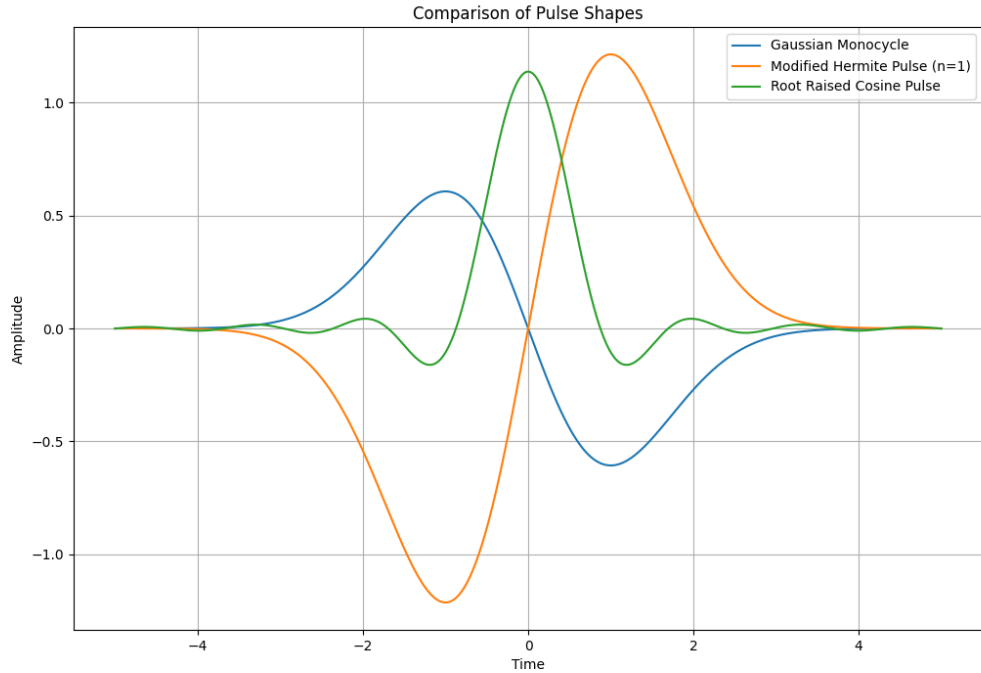


Figure 2.1. Common pulse shapes in ultra-wideband (UWB) systems.

2.1.1.1. Carrier-Based UWB (OFDM-UWB). An alternative to IR-UWB is Orthogonal Frequency Division Multiplexing (OFDM-UWB), which spreads the signal across multiple narrowband carriers within the UWB spectrum. The transmitted OFDM-UWB signal is expressed as

$$s(t) = \sum_{n=1}^N X_n e^{j(2\pi f_n t + \phi_n)}, \quad (2.5)$$

where X_n represents the data symbols, f_n denotes the frequency of the n -th subcarrier, and ϕ_n indicates the phase of the subcarrier. This approach, adopted in standards like IEEE 802.15.3a, enables high data-rate transmission and improved spectral efficiency while leveraging UWB's wide bandwidth. However, the IEEE 802.15.3a task group faced challenges in reaching a consensus between two competing technology proposals: Multi-band Orthogonal Frequency Division Multiplexing (MB-OFDM) and Direct Sequence UWB (DS-UWB). Due to this stalemate, the project was withdrawn in February 2006.

2.1.1.2. Time-Hopping Spread Spectrum (TH-UWB). To mitigate interference and improve multiple-access capability, Time-Hopping UWB (TH-UWB) employs a pseudorandom hopping sequence to vary the pulse position within each frame. The transmitted TH-UWB signal can be expressed as

$$s(t) = \sum_{k=0}^{N-1} a_k p(t - kT_f - c_k T_c), \quad (2.6)$$

where c_k is a pseudorandom integer controlling time shifts, T_c is the chip duration.

This technique enhances multiple access (multi-user communication) and reduces interference with coexisting narrowband systems.

2.1.2. Pulse Modulation and Coding in Ultra-Wideband (UWB) Systems

Ultra-Wideband (UWB) technology primarily employs short-duration pulses instead of continuous wave (CW) carriers for data transmission. These pulses, typically on the order of nanoseconds or sub-nanoseconds, enable high temporal resolution, fine ranging accuracy, and resistance to multipath fading. To efficiently modulate and encode data, various pulse modulation and coding schemes are used in UWB systems.

UWB systems utilize different pulse modulation schemes to encode digital information onto the transmitted pulse waveform. The most common pulse modulation techniques include Pulse Amplitude Modulation (PAM), Pulse Position Modulation (PPM), Binary Phase Shift Keying (BPSK), and Pulse Shape Modulation (PSM).

Pulse Amplitude Modulation (PAM) encodes digital information by varying the amplitude of the transmitted UWB pulse. The transmitted signal can be expressed as

$$s(t) = \sum_{k=0}^{N-1} a_k p(t - kT_f), \quad (2.7)$$

where a_k is the amplitude coefficient representing digital symbols (e.g., $a_k \in \{\pm 1, \pm 3\}$ for 4-PAM), $p(t)$ is the UWB pulse shape, and T_f is the pulse repetition interval. PAM is simple to implement but is highly sensitive to channel noise and multipath fading, making it less favorable for UWB communications.

Pulse Position Modulation (PPM) shifts the timing of each pulse based on the transmitted data. The modulated UWB signal can be written as

$$s(t) = \sum_{k=0}^{N-1} p(t - kT_f - d_k \Delta T), \quad (2.8)$$

where d_k represents the binary data ($d_k \in \{0, 1\}$) and ΔT is the fixed position shift for binary PPM. PPM offers better noise resilience compared to PAM but is sensitive to

timing jitter in practical implementations.

Binary Phase Shift Keying (BPSK) modulates the phase of each pulse according to the input binary data. The transmitted signal is

$$s(t) = \sum_{k=0}^{N-1} p(t - kT_f) \cdot e^{j\pi d_k}, \quad (2.9)$$

where $d_k \in \{0, 1\}$ represents the binary data, and a phase shift of π radians is introduced for "1" symbols. BPSK is widely used in UWB because it provides high spectral efficiency and is robust to noise and multipath effects.

Pulse Shape Modulation (PSM) employs different pulse shapes to represent different symbols. The transmitted signal is:

$$s(t) = \sum_{k=0}^{N-1} p_{d_k}(t - kT_f), \quad (2.10)$$

where $p_{d_k}(t)$ represents the pulse shape corresponding to symbol d_k . PSM is primarily used to control spectral characteristics and enhance interference rejection.

2.1.2.1. Coding Techniques in UWB. To enhance reliability and mitigate interference, UWB systems incorporate error correction coding. In OFDM-based UWB systems, convolutional encoders are typically used, while in Impulse Radio UWB (IR-UWB) systems, Reed-Solomon (RS) encoders are often employed. Additionally, Low-Density Parity-Check (LDPC) codes are used in some UWB implementations to further improve performance under high data-rate conditions and challenging channel environments.

Convolutional Coding: Convolutional coding provides forward error correction (FEC) to protect UWB signals against channel impairments. A rate- R convolutional

encoder processes input bits u_k into coded bits c_k using the formula

$$c_k = \sum_{i=0}^m g_i u_{k-i} \mod 2, \quad (2.11)$$

where g_i are the generator polynomials and m is the memory depth of the encoder. Convolutional coding is typically combined with Viterbi decoding for optimal performance in OFDM-based UWB systems.

Reed-Solomon (RS) Coding: Reed-Solomon coding is a block coding scheme that enhances error resilience in UWB communications. An RS code is denoted as RS(n , k), where n is the total number of symbols in a codeword and k is the number of original information symbols. RS codes are particularly effective in correcting burst errors and are commonly used in IR-UWB systems.

Unlike conventional narrowband signals, UWB signals occupy a significantly larger frequency spectrum, allowing them to coexist with existing wireless systems by operating at very low power levels. This unique characteristic makes UWB technology particularly attractive for precise localization and short-range communication applications [20].

2.1.3. UWB Regulations

Ultra-Wideband (UWB) technology operates under a complex regulatory framework that varies across different regions, balancing innovation with the protection of existing radio services.

Since 2002, the FCC has permitted unlicensed UWB operations in the 3.1 to 10.6 GHz band, with a power spectral density limit of -41.3 dBm/MHz to minimize interference with other services [19].

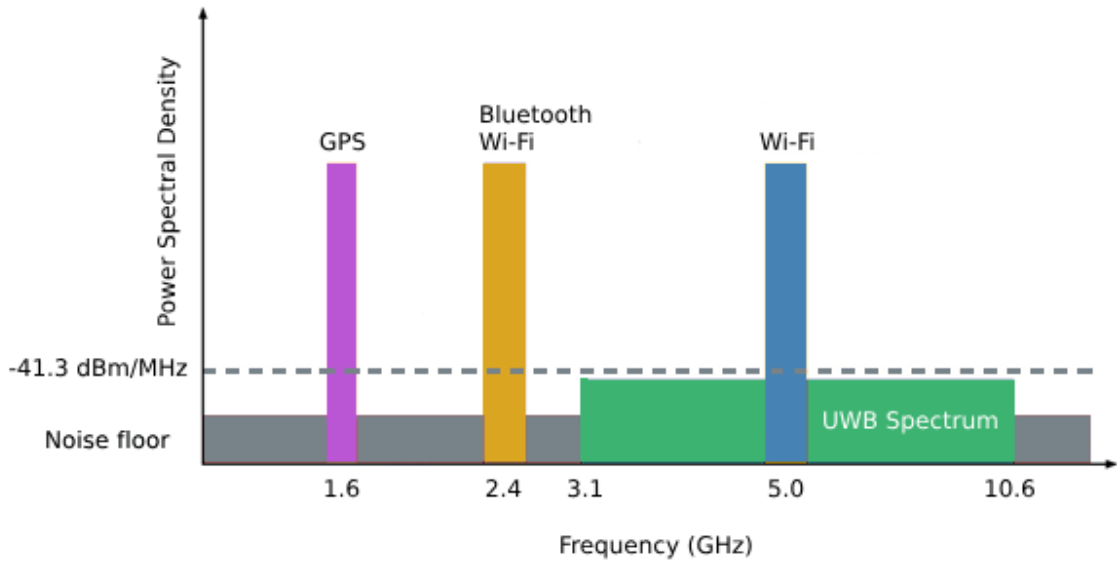


Figure 2.2. UWB -41.3 dBm/MHz limit compared to narrowband standards.

The European Union has recently updated its regulatory stance on UWB. The Commission Implementing Decision (EU) 2024/1467, issued on May 27, 2024, relaxes previous restrictions by allowing higher transmission power levels and permitting outdoor usage in certain frequency channels. These changes are anticipated to enhance the range and applicability of UWB systems, particularly in Real-Time Location Systems (RTLS) [21].

In the Asia-Pacific region, regulatory approaches to UWB vary. Japan and South Korea permit UWB operations primarily in higher frequency channels, such as channel 9 around 8 GHz, to protect existing services in lower bands. Devices are required to implement interference mitigation techniques, especially in the 3.4 to 4.8 GHz range. China is considering regulations that would ban UWB operations in lower frequency channels, aligning with practices in Japan and South Korea. This would standardize the use of higher frequency channels for UWB applications [22].

It is important to note that UWB availability is not uniform worldwide. For

instance, Ultra-Wideband technology is not available in countries such as Armenia, Indonesia, Russia, and several others due to local regulatory restrictions [23].

2.1.4. IEEE Standards for IR-UWB

Ultra-Wideband (UWB) technology offers significant advantages over narrowband systems for localization. Its extremely wide bandwidth provides very high temporal resolution, enabling precise time-of-arrival estimation and excellent multipath discrimination. These properties result in superior ranging accuracy and robust performance in indoor environments, where narrowband systems often struggle due to limited resolution and susceptibility to multipath interference.

IEEE recognized these benefits by standardizing UWB for low-rate wireless personal area networks under IEEE 802.15.4a. This standard defines an Impulse Radio UWB (IR-UWB) physical layer specifically tailored for both communication and high-precision localization. In this context, IR-UWB signals are modeled as a sequence of ultra-short pulses, often generated using a Gaussian monocycle waveform that meets stringent spectral and timing requirements.

For data transmission, the IR-UWB mode in IEEE 802.15.4a employs a hybrid modulation scheme that combines burst position modulation (BPM) with binary phase-shift keying (BPSK). This dual approach encodes information in both the timing of the pulse bursts and in the phase, thereby enhancing resilience against noise and multipath distortions. Moreover, to ensure reliable data delivery under challenging channel conditions, the standard incorporates forward error correction using a systematic Reed-Solomon coding method. This provides additional error protection, improving overall system performance in terms of data integrity and ranging accuracy.

Building on this foundation, IEEE 802.15.4z introduces enhanced coding schemes and refined medium access control (MAC) layer procedures [24]. These advancements further improve the accuracy and security of ranging measurements by offering more

robust error correction and secure, precise time-of-flight ranging. As a result, UWB continues to deliver superior performance in modern localization systems, meeting the evolving requirements of today's applications.

2.2. UWB Market Trends

The global UWB chipset market is projected to grow significantly, with a compound annual growth rate (CAGR) of approximately 19.92% between 2022 and 2030. The market size, valued at around USD 1,478.96 billion in 2022, is expected to reach USD 6,325.89 billion by 2030. This growth is driven by increasing adoption in industries such as consumer electronics, industrial automation, healthcare, and automotive. Leading chipset manufacturers, including Qorvo, NXP Semiconductors, and Decawave (acquired by Qorvo), are actively developing UWB solutions to cater to this expanding demand [25].

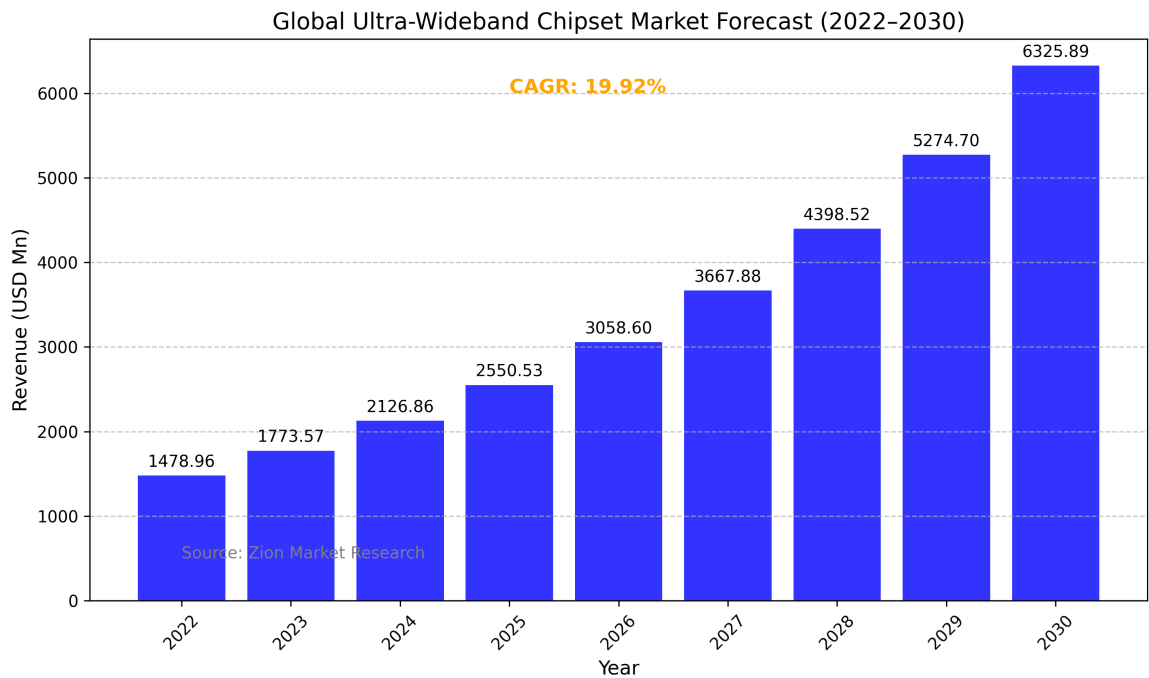


Figure 2.3. Global UWB chipset market forecast (2022-2030).

Organizations like the FiRa Consortium and Omlox are playing a crucial role in standardizing UWB technology for different applications. FiRa Consortium is focused

on ensuring interoperability and enhancing the user experience for UWB-enabled consumer devices, making features like secure access and device-to-device communication more seamless [26]. On the other hand, Omlox is dedicated to industrial applications, aiming to integrate UWB into smart factories, logistics, and other use cases requiring precise real-time localization. These initiatives are accelerating the adoption of UWB across diverse sectors, bridging the gap between research and practical implementation [27].

2.3. Ranging techniques

This section reviews the various ranging techniques utilized in modern localization systems. An overview of their underlying principles, applications, and the challenges associated with achieving accurate positioning is provided before an in-depth discussion of each method.

2.3.1. Two way ranging (TWR)

TWR is a method used to calculate the Time of Flight (ToF) of an RF signal by facilitating bidirectional communication between two devices. The measured ToF is then multiplied by the speed of light to determine the distance between the two devices.

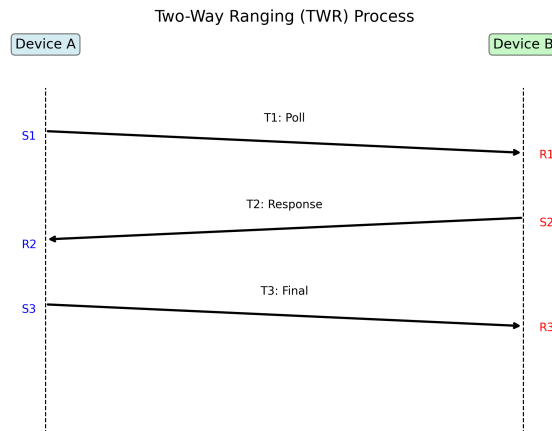


Figure 2.4. DS-TWR process.

In the most commonly used variant, known as Double-Sided Two-Way Ranging (DS-TWR), the process begins with a message called a *poll*, sent by the moving device, referred to as the tag. Upon receiving this message, the fixed device, referred to as the anchor, responds with a message called *response*. Finally, the tag concludes the TWR process by sending a message called *final*, which includes its timestamps. These timestamps are used to calculate the distance with high accuracy.

The calculation of distance involves using the timestamps exchanged during the TWR process. Let S_i denote the transmission timestamp of a message, and R_i denote the reception timestamp. For DS-TWR, the timestamps are defined as

$$S_1, R_1, S_2, R_2, S_3, R_3. \quad (2.12)$$

These timestamps represent the following events during the message exchange process:

- S_1 : The time when the *poll* message is transmitted by the tag.
- R_1 : The time when the *poll* message is received by the anchor.
- S_2 : The time when the *response* message is transmitted by the anchor.
- R_2 : The time when the *response* message is received by the tag.
- S_3 : The time when the *final* message is transmitted by the tag.
- R_3 : The time when the *final* message is received by the anchor.

The Time of Flight (ToF) is computed using the round-trip time formula

$$\text{ToF} = \frac{(R_2 - S_1) - (S_2 - R_1) + (R_3 - S_2) - (S_3 - R_2)}{4}. \quad (2.13)$$

This formula calculates the average time for a signal to travel between the tag and anchor. The distance d between the tag and anchor is then calculated as

$$d = c \cdot \text{ToF}. \quad (2.14)$$

Here, c represents the speed of light in the medium. This method allows for precise ranging without requiring synchronization between the devices.

However, the IEEE 802.15.4a standard allows the usage of clocks with a drift of up to ± 20 ppm. The standard TWR formula does not account for clock drift, which can result in significant errors. To address this issue, an alternative clock drift correction method is proposed in [28]. The corrected Time of Flight (ToF) is computed as

$$\text{ToF} = \frac{(R_2 - S_1)(R_3 - S_2) - (S_3 - R_2)(S_2 - R_1)}{2 \cdot ((R_2 - S_1) + (S_3 - R_2))}. \quad (2.15)$$

Alternatively, it can also be expressed as

$$\text{ToF} = \frac{(R_2 - S_1)(R_3 - S_2) - (S_3 - R_2)(S_2 - R_1)}{2 \cdot ((R_3 - S_2) + (S_2 - R_1))}. \quad (2.16)$$

These formulas eliminate the need for symmetric reply delays while achieving similar accuracy to symmetric double-sided TWR [28]. By minimizing the effects of clock drift, this method ensures reliable ranging even with unsynchronized devices.

2.3.2. Time difference of arrival (TDOA)

TDOA is a localization method that calculates the position of a target device by measuring the time differences at which a signal emitted by the target reaches multiple anchors. Unlike TWR, TDOA requires only one-way communication, where the target device sends a signal that is received by multiple fixed anchors.

In TDOA, each anchor records the timestamp of the signal's arrival. By calculating the differences in these timestamps, the system can derive the relative distances between the target device and the anchors. These differences are then used to determine the target's location by solving a set of hyperbolic equations.

For a 2D localization scenario with three anchors (A_1, A_2, A_3) and a target device

at position (x, y) , the time differences of arrival are given by $\Delta t_{12} = t_1 - t_2$ and $\Delta t_{13} = t_1 - t_3$, where t_1, t_2, t_3 are the signal arrival times at anchors A_1, A_2, A_3 , respectively. The corresponding distances can then be derived as $\Delta d_{12} = c \cdot \Delta t_{12}$ and $\Delta d_{13} = c \cdot \Delta t_{13}$, where c is the speed of light in the medium.

The calculated time differences correspond to hyperbolic curves, where each curve represents the locus of points equidistant from the two anchors. The intersection of these curves provides the precise location of the target.

TDOA is particularly advantageous in systems where power consumption and efficiency are critical. Since the target device only needs to transmit a single signal without participating in multiple communication exchanges, it can operate with lower power compared to TWR. This makes TDOA suitable for applications such as asset tracking and large-scale industrial localization. However, TDOA systems require highly synchronized clocks across the anchors to minimize errors and achieve accurate localization [29].

2.3.3. Angle of Arrival (AoA)

AoA is a localization method that determines the position of a target device by measuring the angle at which its transmitted signal arrives at one or more fixed anchors. This technique relies on specialized antennas or antenna arrays capable of estimating the direction of incoming signals with high precision.

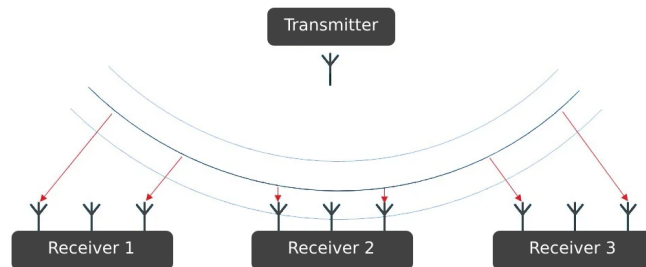


Figure 2.5. AoA estimation.

In an AoA system, when the target device emits a signal, the anchor uses its antenna array to measure the phase differences or time delays between signals received by multiple elements of the array. These phase differences correspond to the angle of arrival of the signal. Mathematically, the angle of arrival (θ) can be computed as

$$\theta = \arcsin\left(\frac{\Delta d}{d_{\text{spacing}}}\right), \quad (2.17)$$

where Δd is the path difference between two antenna elements, and d_{spacing} is the distance between the antenna elements.

To achieve high-accuracy direction estimation, advanced algorithms such as Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) and Multiple Signal Classification (MUSIC) are employed. These algorithms utilize the spatial correlation of signals received by the antenna array to estimate the AoA with sub-degree precision, making them particularly effective in environments with multipath propagation and noise [30].

The AoA measurement provides directional information that, when combined with range measurements (e.g., from TWR or TDOA), enables precise 2D or 3D localization. Using multiple anchors, each with its own AoA estimates, the target's position can be determined through triangulation.

AoA-based systems are particularly advantageous in scenarios requiring high accuracy over shorter distances, such as indoor positioning or asset tracking. However, the accuracy of AoA measurements depends on the quality of the antenna array and its calibration. Environmental factors, such as multipath propagation and signal reflections, can introduce errors in AoA estimation.

2.3.4. Received Signal Strength (RSS)

Received Signal Strength (RSS) is a localization technique that estimates the distance between a transmitter and a receiver by measuring the strength of the received signal. The underlying principle is that the signal strength decreases as the distance between the transmitter and receiver increases, following a predictable path-loss model.

The RSS-based distance estimation can be expressed using the following logarithmic model:

$$P(d) = P(d_0) - 10 \cdot n \cdot \log_{10} \left(\frac{d}{d_0} \right), \quad (2.18)$$

where $P(d)$ is the received signal strength at distance d (in dBm), $P(d_0)$ is the reference signal strength at a known distance d_0 (in dBm), n is the path-loss exponent that depends on the propagation environment, d is the distance between the transmitter and receiver, and d_0 is the reference distance (typically 1 meter in indoor environments).

Using this model, the distance d can be computed by rearranging the equation as

$$d = d_0 \cdot 10^{\frac{P(d_0) - P(d)}{10 \cdot n}}. \quad (2.19)$$

RSS is commonly used in indoor localization systems because it does not require sophisticated hardware like timing or angle-based approaches. However, it suffers from limitations such as multipath fading, shadowing, and variations in the transmission power, which can significantly degrade accuracy in complex environments. To mitigate these issues, fingerprinting methods are often employed. In this approach, a database of signal strengths at known locations is created during an offline phase. During localization, real-time RSS measurements are matched with the database to estimate the location [31].

RSS-based techniques are not widely used in UWB systems due to the high timing resolution that UWB inherently provides, which makes time-based methods like ToF or TDoA more accurate and reliable. Instead, RSS is typically employed in Bluetooth Low Energy (BLE) beacon-based systems, where the expected localization accuracy is generally higher than 5 meters.

2.4. Machine Learning (ML) methods

Machine learning methods used to enhance measurement accuracy and to correct systematic errors in the data are discussed in this section. A brief overview of the range of algorithms employed is provided, laying the foundation for the detailed analysis in the subsequent subsections.

2.4.1. Linear Regression

Linear Regression is a fundamental machine learning algorithm used for modeling the relationship between a dependent variable y and one or more independent variables x . It assumes a linear relationship between the variables and aims to find the best-fitting line that minimizes the prediction error. For a single-variable case (simple linear regression), the model is represented as

$$y = \beta_0 + \beta_1 x + \epsilon, \quad (2.20)$$

where y is the dependent variable (response), x is the independent variable (predictor), β_0 is the intercept of the regression line, β_1 is the slope of the regression line, and ϵ is the error term, accounting for variability not explained by the model.

The parameters β_0 and β_1 are estimated using the method of least squares, which minimizes the sum of squared residuals (RSS) between the observed values and the predicted values. The residuals are defined as the differences between the actual values

y_i and the predicted values $\hat{y}_i$:

$$\text{RSS} = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2. \quad (2.21)$$

In the case of multiple independent variables (multiple linear regression), the model generalizes to

$$y = \beta_0 + \sum_{j=1}^p \beta_j x_j + \epsilon, \quad (2.22)$$

where p is the number of predictors.

Linear Regression is widely used in various applications, including localization systems, where it can model relationships between measured data (e.g., distances or signal strengths) and actual positions. However, its simplicity can limit its performance in capturing complex, nonlinear relationships. Despite its limitations, Linear Regression is computationally efficient and provides an interpretable model, making it a popular choice for baseline models in machine learning pipelines.

2.4.2. Polynomial Regression

Polynomial Regression is an extension of Linear Regression that models the relationship between the dependent variable y and the independent variable x as a polynomial of degree n . This method is particularly useful for capturing non-linear relationships between variables while maintaining the simplicity of a regression framework.

The model for Polynomial Regression is given by

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon, \quad (2.23)$$

where y is the dependent variable (response), x is the independent variable (predictor),

$\beta_0, \beta_1, \dots, \beta_n$ are the model coefficients, n is the degree of the polynomial, and ϵ is the error term accounting for variability not explained by the model.

In practice, Polynomial Regression is implemented by transforming the original input features into polynomial features. For example, a dataset with a single feature x would be expanded to include $x, x^2, x^3, \dots, x^n$. Linear Regression is then applied to these transformed features to estimate the coefficients $\beta_0, \beta_1, \dots, \beta_n$.

The degree n of the polynomial controls the flexibility of the model. A higher degree allows the model to capture more complex relationships in the data but may lead to overfitting, where the model fits the training data too closely and fails to generalize to new data. Conversely, a lower degree may result in underfitting, where the model is too simplistic to capture the underlying patterns.

Polynomial Regression is commonly used in applications such as trajectory prediction, sensor data modeling, and environmental data analysis. While it provides a simple way to handle non-linear relationships, its performance may degrade in high-dimensional spaces or when the relationship between variables is highly complex. In such cases, more sophisticated non-linear models, such as neural networks, may be more effective.

2.4.3. Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a type of feedforward artificial neural network widely used for supervised learning tasks such as classification and regression. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of neurons that apply a weighted sum followed by a non-linear activation function.

For a single neuron, the output is given by

$$z = \phi \left(\sum_{i=1}^n w_i x_i + b \right), \quad (2.24)$$

where w_i is the weight for the i -th input, x_i is the i -th input feature, b is the bias term, and ϕ is the activation function (e.g., ReLU, sigmoid, tanh).

The training of an MLP involves minimizing a loss function (e.g., cross-entropy for classification, mean squared error for regression) using backpropagation and gradient descent. The weights are updated iteratively to reduce the loss by applying the following weight update rule:

$$w_i \leftarrow w_i - \eta \frac{\partial L}{\partial w_i}, \quad (2.25)$$

where η is the learning rate and L is the loss function.

MLPs are capable of modeling complex non-linear relationships and are widely used in applications such as speech recognition, image classification, and sensor data analysis. However, their performance depends on the choice of hyperparameters (e.g., number of layers, neurons, and learning rate) and requires substantial computational resources for training on large datasets.

2.4.4. Alternative Machine Learning Methods

Several alternative machine learning algorithms are widely used for tasks similar to UWB-based localization, including:

- **Support Vector Machine (SVM):** Effective for classification tasks with small datasets but computationally expensive for large-scale problems.
- **Decision Tree:** An interpretable method for classification and regression but prone to overfitting on complex datasets.

- **Random Forest:** An ensemble learning method that reduces overfitting and increases accuracy but lacks the interpretability of simpler models.

While these methods have their merits, this work focuses on **Multilayer Perceptron (MLP)** due to its demonstrated effectiveness in UWB-based localization systems. Previous studies [32–34] have evaluated the performance of MLP in this domain, showcasing its ability to achieve high accuracy for tasks such as NLOS (Non-Line of Sight)/LOS (Line of Sight) identification, mitigating environmental interference, and improving overall localization precision. Based on these findings, MLP was selected as the primary machine learning model in this research.

2.5. Localization Algorithms

This section presents an in-depth examination of the localization algorithms that underpin the study’s positioning system. Basic principles of multilateration are outlined, and a comparison of several alternative methods is presented in the subsections that follow.

2.5.1. Multilateration

Multilateration is a localization technique that determines the position of a target by measuring its distances from multiple reference points, often referred to as anchors, with known coordinates. It relies on the principle that the target must lie on the surface of a sphere centered at each anchor, with the radius corresponding to the measured distance [35].

In a 2D scenario, the target’s position is determined by the intersection of circles centered at the anchors, while in a 3D scenario, spheres are used instead. Let (x_T, y_T, z_T) represent the coordinates of the target, and (x_i, y_i, z_i) represent the known coordinates of the i -th anchor. The measured distance d_i between the target and the

i -th anchor is given by

$$d_i = \sqrt{(x_T - x_i)^2 + (y_T - y_i)^2 + (z_T - z_i)^2}, \quad (2.26)$$

where d_i is the Euclidean distance.

In practical implementations, measurement noise and inaccuracies prevent perfect intersections of these geometric shapes. To address this, multilateration employs optimization techniques to estimate the most likely position of the target by minimizing the residual error between the measured and calculated distances using the following least squares objective function

$$\min_{x_T, y_T, z_T} \sum_{i=1}^n \left(\sqrt{(x_T - x_i)^2 + (y_T - y_i)^2 + (z_T - z_i)^2} - d_i \right)^2. \quad (2.27)$$

The least squares method is commonly used to solve this optimization problem. By linearizing the non-linear distance equations through techniques such as Taylor series expansion, the problem is reformulated as a system of linear equations to

$$Ax = b, \quad (2.28)$$

where A is a matrix representing the anchor positions, and b is a vector derived from the measured distances. The optimal position x is then obtained by solving

$$x = (A^T A)^{-1} A^T b. \quad (2.29)$$

This approach provides an efficient and robust solution for multilateration, particularly in environments with moderate measurement noise.

Multilateration is commonly used in systems such as UWB localization and radar systems, where precise distance measurements are available. A key advantage of mul-

tilateralation is its ability to operate passively, requiring the target to only respond to signals rather than initiate communication. This property makes it suitable for applications like air traffic control and UAV localization.

However, the accuracy of multilateration depends on several factors, including:

- **Geometry of the anchors:** Poorly placed anchors can result in large geometric dilution of precision (GDOP), reducing accuracy.
- **Quality of the measurements:** Noise and environmental interference can significantly impact the distance measurements.
- **Number of anchors:** A minimum of three anchors is required for 2D localization, while four anchors are needed for 3D localization.

2.5.2. Kalman Filter

The Kalman Filter (KF) is an optimal recursive algorithm used for estimating the state of a dynamic system in real-time. It combines noisy measurements with a system model to provide estimates that minimize the mean squared error. Due to its efficiency and simplicity, the Kalman Filter is widely used in applications such as navigation, localization, and sensor fusion.

The Kalman Filter operates on a discrete-time linear system, which is typically represented as

$$x_k = Ax_{k-1} + Bu_k + w_k, \quad (2.30)$$

$$z_k = Hx_k + v_k, \quad (2.31)$$

where x_k is the state vector at time step k (e.g., position and velocity), A is the state transition matrix, B is the control input matrix, u_k is the control input vector at time

k , w_k is the process noise assumed to be zero-mean Gaussian with covariance Q , z_k is the measurement vector at time k (e.g., sensor readings), H is the observation matrix, and v_k is the measurement noise assumed to be zero-mean Gaussian with covariance R .

The Kalman Filter consists of two main steps:

(i) **Prediction Step:**

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k, \quad (2.32)$$

$$P_k^- = AP_{k-1}A^T + Q, \quad (2.33)$$

where $\hat{x}_k^-$ is the predicted state, and P_k^- is the predicted error covariance.

(ii) **Update Step:**

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}, \quad (2.34)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-), \quad (2.35)$$

$$P_k = (I - K_k H)P_k^-, \quad (2.36)$$

where K_k is the Kalman gain, $\hat{x}_k$ is the updated state estimate, and P_k is the updated error covariance.

The Kalman Filter is particularly effective in scenarios where real-time estimation is required, as it updates the state estimates sequentially with each new measurement. However, it assumes that the system is linear and that the noise is Gaussian, which

can limit its applicability in non-linear or non-Gaussian environments.

To address these limitations, variants such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) have been developed. These methods extend the Kalman Filter's principles to handle non-linear systems by approximating the non-linearities in different ways.

2.5.3. Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF) is a variant of the Kalman Filter designed to handle non-linear dynamic systems. While the standard Kalman Filter assumes linear models, the EKF extends its applicability by linearizing the non-linear system around the current state estimate. This makes it suitable for a wide range of applications, including robotics, navigation, and localization, where system dynamics are inherently non-linear.

A non-linear system is typically represented as:

$$x_k = f(x_{k-1}, u_k) + w_k, \quad (2.37)$$

$$z_k = h(x_k) + v_k, \quad (2.38)$$

where x_k is the state vector at time step k , $f(x_{k-1}, u_k)$ is the non-linear state transition function, u_k is the control input vector, w_k is the process noise assumed to be zero-mean Gaussian with covariance Q , z_k is the measurement vector at time k , $h(x_k)$ is the non-linear observation function, and v_k is the measurement noise assumed to be zero-mean Gaussian with covariance R .

The EKF algorithm follows two main steps:

(i) **Prediction Step:**

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k), \quad (2.39)$$

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q, \quad (2.40)$$

where $\hat{x}_k^-$ is the predicted state, and P_k^- is the predicted error covariance. The Jacobian matrix $F_{k-1} = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}_{k-1}}$ linearizes the state transition function around $\hat{x}_{k-1}$.

(ii) **Update Step:**

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}, \quad (2.41)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)), \quad (2.42)$$

$$P_k = (I - K_k H_k) P_k^-, \quad (2.43)$$

where K_k is the Kalman gain, $\hat{x}_k$ is the updated state estimate, and P_k is the updated error covariance. The Jacobian matrix $H_k = \frac{\partial h}{\partial x} \Big|_{x=\hat{x}_k^-}$ linearizes the observation function around $\hat{x}_k^-$.

The EKF's ability to handle non-linear systems makes it a valuable tool for UWB-based localization, where dynamic motion and sensor fusion introduce non-linearities. By combining UWB range measurements with data from other sensors, such as IMUs, the EKF provides accurate position and velocity estimates even in complex environments.

2.5.4. Alternative Localization Algorithms

While the Extended Kalman Filter (EKF) is widely used for state estimation in dynamic systems, alternative localization algorithms such as the Unscented Kalman Filter (UKF) and Particle Filters offer enhanced performance in certain scenarios.

Unscented Kalman Filter (UKF): The UKF addresses some of the limitations of the EKF by avoiding linearization through Jacobians. Instead, it uses a deterministic sampling approach known as the Unscented Transform to approximate the mean and covariance of the system's state. This allows the UKF to handle highly non-linear systems more accurately than the EKF. However, the computational overhead of generating and propagating sigma points is significantly higher, making it less suitable for real-time applications with strict timing requirements.

Particle Filter: Particle Filters are non-parametric Bayesian methods that approximate the posterior distribution of the system state using a set of weighted particles. They can model arbitrary non-linearities and non-Gaussian noise, offering superior accuracy compared to both EKF and UKF in complex systems. However, the computational cost of maintaining and updating a large number of particles can be prohibitively high, especially for real-time applications.

Despite the advantages of these algorithms, their computational complexity poses challenges for real-time performance. In the context of this thesis, achieving real-time localization is critical for UAV operations. While UKF and Particle Filters might provide better accuracy in certain cases, their higher computational requirements make them less suitable for this application. Therefore, the Extended Kalman Filter (EKF) is selected as the localization algorithm due to its balance between computational efficiency and accuracy, meeting the real-time performance requirements of the system.

3. METHODOLOGY

This chapter outlines the methodology for UWB-based drone localization during autonomous landings. A detailed description of the hardware, calibration, and real-time data processing procedures is provided to achieve high-precision positioning in GNSS-denied environments.

3.1. System Overview

The proposed localization system is designed to provide high-accuracy UAV positioning for autonomous landing in GNSS-denied environments. The system is built around a $0.6\text{ m} \times 0.7\text{ m}$ rectangular base on which four UWB anchors are placed at the corners. All UWB hardware is based on the DWM1001-DEV development kit, produced by Qorvo, offering reliable and precise ranging capabilities [36].

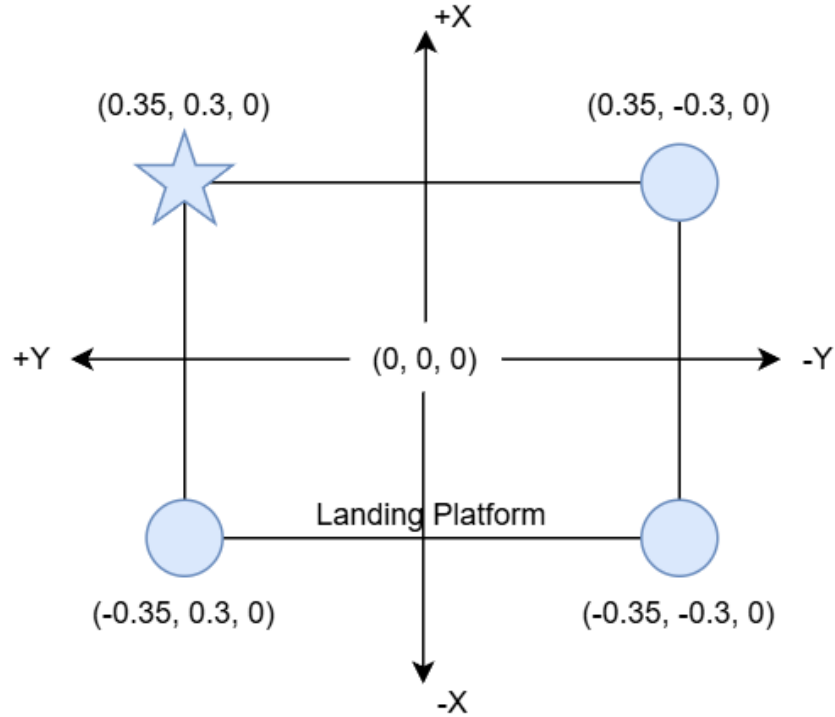


Figure 3.1. Anchor coordinates and positioning coordinate system.

To enable orientation estimation, one of the anchors and the UWB tag have been modified by integrating additional sensors. These modifications enable both the anchor and the tag to compute their orientation relative to the East-North-Up (ENU) reference frame using the VQF orientation estimation algorithm. Figure 3.1 illustrates the positioning coordinate system and the locations of the UWB anchors.

The modified anchor is positioned at the coordinates $(0.35, 0.3, 0)$ and its integrated IMU sensor is aligned with the coordinate system. This alignment ensures that the orientation data is directly consistent with the reference frame, eliminating the need for additional rotation of the acceleration vector during data processing. Such alignment simplifies the computation and improves the accuracy of orientation integration within the localization system.

All data collected during each TWR session flows to a Raspberry Pi 5, which serves as the central processing unit. The Raspberry Pi powers all UWB anchors via its USB bus, streamlining the hardware configuration and providing an efficient power solution. A C-based application running on the Raspberry Pi handles:

- Data acquisition from the UWB anchors and sensors.
- Preprocessing of raw data to reduce noise and handle outliers.
- Real-time computation of the UAV's position using an EKF.

The EKF fuses the corrected UWB ranging data with orientation information from the modified tag and anchor to compute accurate localization estimates. The system achieves a high update rate of 10 ms per TWR session, ensuring real-time performance suitable for dynamic UAV operations.

This architecture leverages the computational capabilities of the Raspberry Pi 5, enabling efficient multi-threaded data collection and real-time processing. The system is designed to provide robust and reliable localization, meeting the requirements for autonomous UAV landings.

3.2. Hardware Components

This section describes the key hardware components that enable the UWB-based localization system, detailing their configuration and role in achieving high-accuracy positioning.

3.2.1. UWB transceiver and microcontroller

Figure 3.2 illustrates the DWM1001-DEV device, one of the primary components used in the system. This device, produced by Qorvo, serves as a development kit for the DWM1001 module.



Figure 3.2. DWM1001-DEV from Qorvo.

The internal structure of the DWM1001 module is shown in Figure 3.3. At its core, the DWM1001 module uses the NRF52832 System-on-Chip (SoC) from Nordic Semiconductor as its microcontroller unit (MCU). The NRF52832 features a 64 MHz Arm Cortex-M4F processor, 64 kB of RAM, and 512 kB of flash memory [37]. Additionally, the SoC includes an integrated BLE transceiver with support for BLE 5.4 protocols, utilizing softdevice firmware.

The DWM1001 module incorporates Qorvo’s DW1000 transceiver as the UWB transceiver [38, 39]. The DW1000 transceiver is connected to the NRF52832 via an SPI bus within the module. The SPI bus is configured to operate at the maximum clock speed supported by the NRF52832, 8 MHz, minimizing the overhead caused

by SPI transactions. A custom-designed PCB antenna, manufactured by Qorvo, is connected to the DW1000 transceiver. This omnidirectional antenna is optimized for IEEE 802.15.4a channel 5, covering the frequency range of 6240 MHz to 6739.2 MHz.

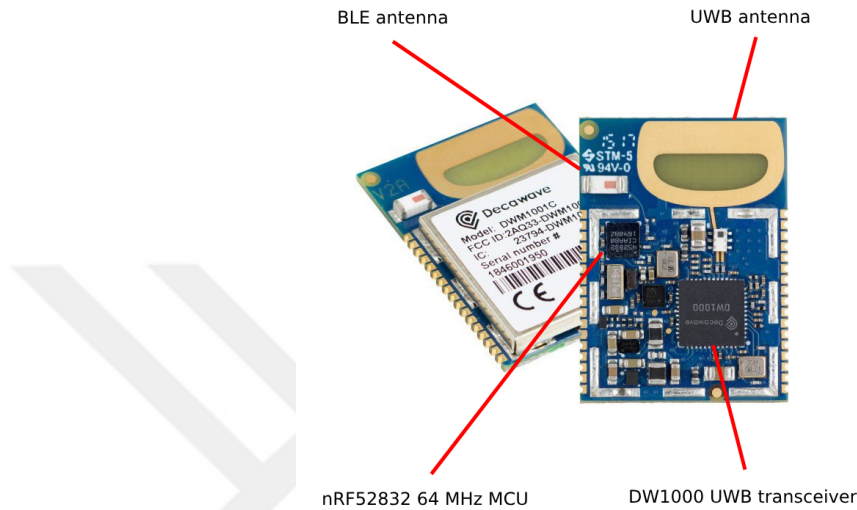


Figure 3.3. DWM1001 module internal structure.

In addition to the MCU and UWB transceiver, the DWM1001 module integrates an LIS2DH12 accelerometer from STMicroelectronics. This accelerometer provides motion data, further enhancing the module’s capabilities for localization and dynamic applications.

The DWM1001-DEV development board features a USB connection and an on-board J-Link debugger, which collectively support reprogramming, debugging, and powering the device. These functionalities simplify the development process by providing a unified interface for code deployment, troubleshooting, and power management.

In this work, the DW1000 device driver was utilized in the development of the firmware for the UWB transceiver. The device driver, provided by Decawave, offers a comprehensive set of API functions for initializing, configuring, transmitting, and receiving data with the DW1000. Written in C and designed for portability across various microcontroller platforms, it facilitates efficient control of the transceiver and

enables high-precision ranging measurements [40].

Furthermore, in sections where direct register access is required for fine-tuned control of the transceiver, the DW1000 User Manual was used as the primary reference. This manual provides detailed descriptions of the internal register map, default values, and operational parameters of the DW1000 [41].



Figure 3.4. UWB tag hardware.

Figure 3.4 illustrates the UWB tag hardware. The DWM1001-DEV PCB was removed from its original enclosure and integrated into a custom-designed enclosure to house all the required components in a single unit. A 2900 mAh 18650 Li-Ion battery was added to power the system, along with a Li-Ion battery charger to facilitate recharging.

Additionally, a 9 DoF IMU module (GY-85 PCB) was integrated into the enclosure to provide orientation and motion data. To support the operation of a laser sensor, the tag includes a 3.0-4.2V to 5.0V boost converter for voltage regulation. This compact design ensures that all components are securely housed and properly powered, making the tag suitable for dynamic UAV applications.

Overall weight of the UWB tag is approximately 161 grams, with dimensions of

13 cm $\times$ 8 cm $\times$ 3.3 cm. The tag is designed to be lightweight and compact, ensuring minimal impact on the UAV's payload while providing essential localization capabilities.

The UWB anchor customized hardware is similar to the UWB tag, with the primary difference being the absence of the laser sensor. The battery is included but is not required for operation, as the anchor can be powered via USB.

3.2.2. Inertial Measurement Unit (IMU)

Accurate orientation estimation is a critical component for UAV localization. Without using a magnetometer, absolute orientation cannot be determined. When relying solely on 6D sensor fusion using an accelerometer and a gyroscope, only local orientation can be estimated, which lacks a reference to true north or an absolute frame. Furthermore, such systems are prone to drift, with inaccuracies accumulating at a rate of up to a few degrees per second due to sensor inaccuracies. To address this limitation, the GY-85 board was included in the system, providing a combination of accelerometer, gyroscope, and magnetometer sensors.

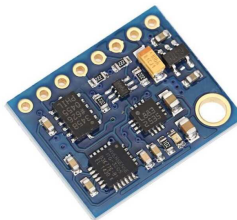


Figure 3.5. GY-85 sensor board.

The GY-85 board integrates the following components:

- **ADXL345 Accelerometer:** A 3-axis accelerometer from Analog Devices, capable of high-resolution (13-bit) measurements up to $\pm 16g$ [42]. It is designed

for dynamic and static acceleration sensing, ensuring precise measurements even under rapid movements.

- **ITG3205 Gyroscope:** A digital-output X-, Y-, and Z-axis angular rate sensor from TDK InvenSense. The gyroscope has a sensitivity of 14.375 LSBs per °/sec and a full-scale range of $\pm 2000^\circ/\text{sec}$, enabling it to capture high-speed rotations with accuracy [43].
- **HMC5883L Magnetometer:** A high-resolution, low-field magnetic sensing module from Honeywell. This surface-mount device features a 12-bit ADC for precise magnetic field measurements, allowing 1° to 2° compass heading accuracy [44]. It includes advanced features such as amplification, automatic degaussing strap drivers, offset cancellation, and high-resolution magneto-resistive sensors, making it ideal for low-cost compassing and magnetometry applications.

All sensors on the GY-85 board are powered by a 3V power source supplied by the DWM1001-DEV PCB. The board is connected to the system via an I²C bus, ensuring seamless integration and efficient communication. By leveraging the GY-85 board's sensors, the system achieves robust and accurate orientation estimation, minimizing drift and providing absolute orientation with respect to the Earth's magnetic field.

3.2.3. Laser Sensor

To provide accurate altitude measurements for the UAV, a laser range sensor was integrated into the system. This altitude data is critical for maintaining precise vertical positioning during autonomous landing. For this purpose, the Benewake TFmini Plus sensor was selected, offering reliable distance measurement in a compact and lightweight form factor.

The key specifications of the TFmini Plus are as follows [45]:

- **Measurement Range:** 0.1 to 12 meters.
- **Accuracy:** ± 6 cm (for distances < 6 m) and $\pm 1\%$ (for distances ≥ 6 m).

- **Frame Rate:** Up to 1000 Hz, enabling high-frequency updates suitable for real-time applications.
- **Operating Voltage:** 5V.
- **Interface:** Supports UART and I²C communication protocols, allowing seamless integration with the DWM1001-DEV PCB.
- **Compact Design:** Dimensions of 42 mm × 15 mm × 16 mm and a weight of only 11 grams, minimizing the impact on the UAV's payload.



Figure 3.6. Benewake TFmini Plus laser sensor.

The TFmini Plus was chosen for its balance of high accuracy, fast response time, and ease of integration. Its measurement range and precision make it well-suited for UAV altitude determination, even in dynamic environments. The sensor communicates with the system via its I²C interface and is powered by a 5V boost converter included in the UWB tag's enclosure. By incorporating this laser sensor, the system achieves reliable and precise altitude measurements, enhancing the overall localization accuracy.

3.2.4. Raspberry Pi 5 Single Board Computer

The Raspberry Pi 5 plays a critical role in the system, serving as the central processing unit that manages data acquisition, preprocessing, and real-time localization. With its advanced hardware capabilities, the Raspberry Pi 5 is uniquely suited to handle the intensive computational demands of UAV localization while maintaining compactness and energy efficiency.

At its core, the Raspberry Pi 5 is powered by a quad-core Arm Cortex-A76 CPU running at 2.4 GHz, providing a substantial performance boost over previous iterations [46]. This enhanced processing power ensures that complex tasks, such as EKF calculations and real-time data fusion, are executed with high efficiency. Paired with options for either 4 GB or 8 GB of LPDDR4X RAM, the system can seamlessly handle multi-threaded operations, enabling simultaneous data collection and processing from multiple sensors without performance bottlenecks.

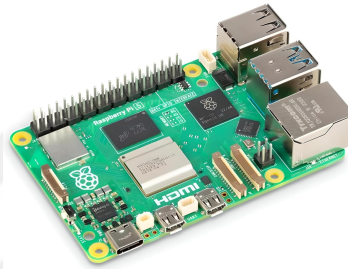


Figure 3.7. Raspberry Pi 5 single board computer.

In addition to its computational capabilities, the Raspberry Pi 5 integrates robust connectivity and storage options. The system is equipped with multiple USB 3.0 and USB 2.0 ports, which provide high-speed data transfer from connected UWB anchors. The USB-C power input not only powers the Raspberry Pi but also supplies consistent and reliable power to all four UWB anchors via the USB bus, streamlining the hardware configuration. For storage, the Raspberry Pi supports high-speed microSD cards and an optional eMMC module, ensuring fast access to the system's data and applications.

Its support for Gigabit Ethernet and Wi-Fi 6 ensures that the system remains adaptable for future expansions, such as cloud-based data processing or communication with other UAV systems.

The Raspberry Pi 5 runs a C-based multi-threaded application specifically developed for this system. This application collects data from UWB anchors which include distance measurements, acceleration and current orientations in real-time, preprocesses the measurements to filter out noise and reject outliers, and performs localization calcu-

lations using the EKF. The computed position and orientation data are then reported back to the UAV with near real-time responsiveness, ensuring the system's suitability for dynamic UAV operations.

With its powerful processor, advanced connectivity, and compact form factor, the Raspberry Pi 5 not only meets the computational demands of the system but also satisfies the constraints of size, weight, and power that are critical for UAV applications. Its ability to handle real-time processing while maintaining energy efficiency makes it an indispensable component in achieving precise and reliable UAV localization.

3.3. Calibration

3.3.1. Inertial Measurement Unit (IMU) Calibration

Calibrating each sensor in the IMU is essential to ensure accurate orientation estimation. Without proper calibration, raw sensor readings are prone to significant inaccuracies caused by manufacturing imperfections, environmental interference, and mounting misalignments.

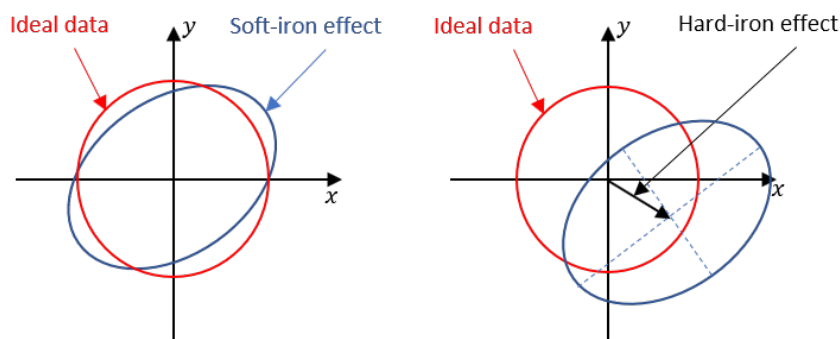


Figure 3.8. Hard iron and soft iron effects on magnetometer readings.

One critical aspect of IMU calibration is the magnetometer, which provides absolute orientation by measuring the Earth's magnetic field. Ideally, when magnetometer readings are taken from all angles, they should form a perfect sphere in a three-dimensional space. However, due to sensor inaccuracies, the data often forms an

ellipsoid instead. These inaccuracies can be attributed to hard iron effects (permanent magnetic interference) and soft iron effects (distortions caused by nearby ferromagnetic materials).

To correct these distortions, a program called Magneto was used. Magneto computes the optimal soft iron and hard iron calibration parameters based on data collected from the magnetometer. Using Li's least squares ellipsoid-specific fitting algorithm, the program determines the following [47]:

- $\mathbf{A}^{-1}$: The soft iron correction matrix.
- $\mathbf{b}$: The hard iron offset vector.

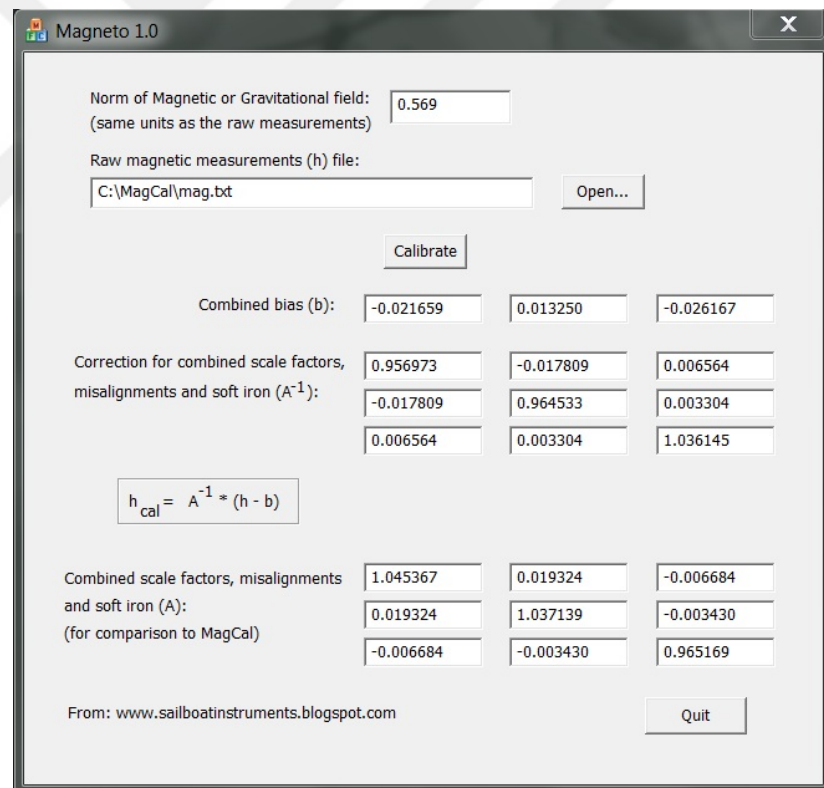


Figure 3.9. Magneto calibration software.

The calibrated magnetometer measurements, $\mathbf{h}_{\text{cal}}$, are computed to correct for distortions and offsets in the raw magnetometer readings. The calibration is performed

using the formula:

$$\mathbf{h}_{\text{cal}} = \mathbf{A}^{-1} \cdot (\mathbf{h} - \mathbf{b}), \quad (3.1)$$

where $\mathbf{h}$ denotes the raw magnetometer readings, $\mathbf{A}^{-1}$ is the inverse of the calibration matrix that compensates for soft iron distortions, and $\mathbf{b}$ is the bias vector that accounts for hard iron offsets.

This calibration process ensures that the magnetometer provides accurate and reliable measurements, enabling precise orientation estimation.

The accelerometer also requires calibration to ensure accurate readings. When held perpendicularly to any axis, the accelerometer does not always register exactly 1g. Additionally, even when perfectly aligned with one axis, slight accelerations are often measured on other axes due to cross-axis effects. To address these inaccuracies, a similar principle was applied to calibrate the accelerometer using the Magneto program. During the calibration process, data was collected in six directions: $+X$, $-X$, $+Y$, $-Y$, $+Z$, and $-Z$, with the accelerometer held perpendicularly in each orientation. The collected data was input into the Magneto program, which provided:

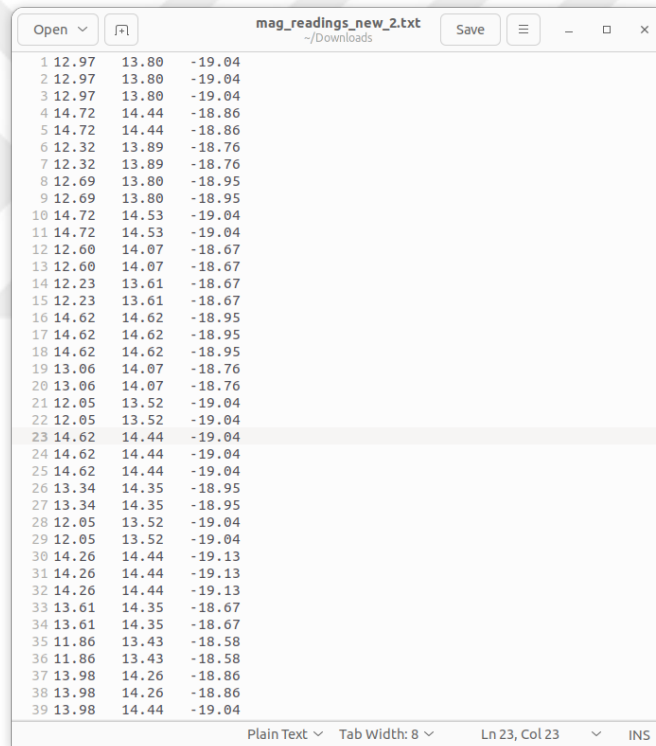
- $\mathbf{A}^{-1}$: The matrix for compensating cross-axis effects and scaling the measurements.
- $\mathbf{b}$: The vector representing biases for each axis.

Prior to performing accelerometer or magnetometer calibration, sensor data was acquired using a simple firmware flashed onto the DWM1001 module. In this firmware, the sensor is configured to generate an interrupt whenever a DATAREADY event occurs. The microcontroller (MCU) handles this interrupt by reading the sensor values, converting them into physical units, and sending the values over UART in the following

format:

"Value_x\tValue_y\tValue_z\n"

Here, \t represents a horizontal tab, and \n indicates a new line. Since the board includes a USB-UART bridge, it is connected to a PC via a USB cable. A serial terminal program, such as TeraTerm or PuTTY, captures the output and saves it into a .txt file. This file is then provided to the Magneto program to calculate the optimal calibration values.



Line	Value _x	Value _y	Value _z
1	12.97	13.80	-19.04
2	12.97	13.80	-19.04
3	12.97	13.80	-19.04
4	14.72	14.44	-18.86
5	14.72	14.44	-18.86
6	12.32	13.89	-18.76
7	12.32	13.89	-18.76
8	12.69	13.80	-18.95
9	12.69	13.80	-18.95
10	14.72	14.53	-19.04
11	14.72	14.53	-19.04
12	12.60	14.07	-18.67
13	12.60	14.07	-18.67
14	12.23	13.61	-18.67
15	12.23	13.61	-18.67
16	14.62	14.62	-18.95
17	14.62	14.62	-18.95
18	14.62	14.62	-18.95
19	13.06	14.07	-18.76
20	13.06	14.07	-18.76
21	12.05	13.52	-19.04
22	12.05	13.52	-19.04
23	14.62	14.44	-19.04
24	14.62	14.44	-19.04
25	14.62	14.44	-19.04
26	13.34	14.35	-18.95
27	13.34	14.35	-18.95
28	12.05	13.52	-19.04
29	12.05	13.52	-19.04
30	14.26	14.44	-19.13
31	14.26	14.44	-19.13
32	14.26	14.44	-19.13
33	13.61	14.35	-18.67
34	13.61	14.35	-18.67
35	11.86	13.43	-18.58
36	11.86	13.43	-18.58
37	13.98	14.26	-18.86
38	13.98	14.26	-18.86
39	13.98	14.44	-19.04

Figure 3.10. Contents of magnetometer reading file as example.

The gyroscope was also calibrated by taking 10,000 samples while the device was stationary. These samples were averaged to determine the biases for each axis, which were then used to correct subsequent measurements. Soft-iron-like calibration for the gyroscope presents additional challenges, as it requires the device to be rotated at a known angular speed for each axis, making the process more complex. As a result,

only bias calibration was performed for the gyroscope in this system.

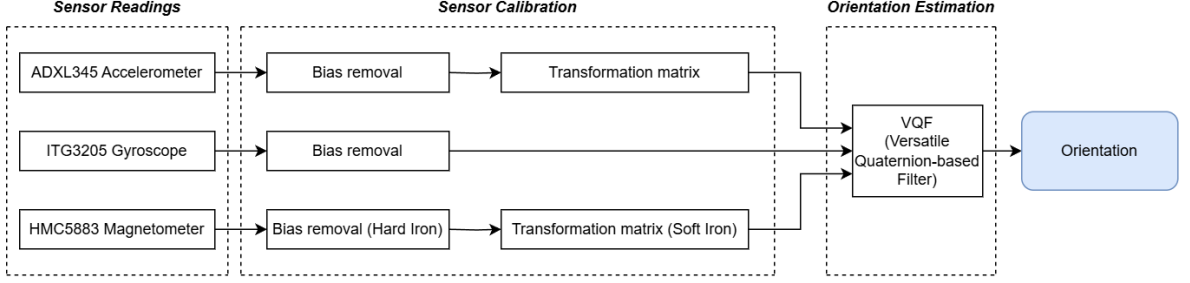


Figure 3.11. Orientation estimation using the calibrated IMU.

This comprehensive calibration process ensures that the IMU sensors provide accurate and reliable data, enabling robust orientation estimation and minimizing drift in the system.

3.3.2. UWB Calibration

Calibration of UWB systems is critical to ensure high-accuracy ranging, as both antenna delay and ranging bias significantly influence performance.

Antenna Delay Calibration: Antenna delay refers to the internal propagation delay within the DW1000 transceiver. This delay includes the time taken for signals to propagate through the transmitter and receiver paths of the device [48]. The total measured time can be expressed as

$$t_{\text{measured}} = t_{\text{ADTX}} + \text{ToF} + t_{\text{ADRX}}, \quad (3.2)$$

where t_{ADTX} and t_{ADRX} are the transmit and receive antenna delays, respectively. Although these delays are not part of the Time of Flight (ToF) measurement itself, they contribute to the total delay and must be accounted for to extract accurate ToF. To address this, DWM1001 devices undergo antenna delay calibration at the factory. The calibration values are written into the one-time programmable (OTP) memory of the DW1000 transceiver, ensuring device-level consistency. These values were read from the

OTP memory using the DW1000 device driver API function `dwt_otpread` and applied using the `dwt_setrxantennadelay` and `dwt_settxantennadelay` functions. Utilizing these calibrated values minimizes the impact of antenna delays on ranging accuracy, thereby enhancing the overall precision of the UWB system.

Clock Drift Calibration: Clock drift is another factor that contributes to ranging errors in UWB systems. Variations in clock frequencies between devices result in inaccuracies during timestamp calculations. To mitigate this, the DW1000 transceiver allows for clock trimming adjustments with a resolution of ± 1.5 ppm steps. Furthermore, DWM1001C modules are factory-calibrated against a reference clock to minimize the impact of clock drift. DW1000 API function `dwt_initialize` reads the calibration values from the OTP memory and applies them to the device. This calibration significantly reduces the error introduced by timing discrepancies, enhancing the overall ranging precision [49].

Ranging Bias: Ranging bias arises due to the dependency of timestamp accuracy on the received signal level (RSL). However, the RSL approximation provided by the DW1000 transceiver is not sufficiently accurate for precise bias correction. To address this, many applications rely on creating a lookup table of measured distances and corresponding offsets. Lookup tables are generated by empirically measuring distances and associating them with appropriate corrections, which are applied during runtime to improve accuracy.

In this work, rather than relying on a conventional lookup table, a machine learning-based approach was adopted to correct UWB ranging errors. Specifically, a MLP model was trained using raw distance and diagnostic features extracted from the UWB system. Notably, a significant portion of these features are the channel impulse response (CIR) metrics.

The CIR characterizes the channel's response to a transmitted impulse, providing a detailed, time-resolved representation of the multipath propagation environment. In

DW1000 based systems, the CIR is measured with extremely high temporal resolution (approximately 1 nanosecond), capturing the precise time delays and amplitudes of individual multipath components such as reflections, diffractions, and scattering. The DW1000 transceiver achieves this through an accumulator that records complex values for each tap. Each tap consists of a 16-bit real integer and a 16-bit imaginary integer, with each tap representing a 1 ns sample interval (equivalent to half a period of the 499.2 MHz fundamental frequency). The entire accumulator spans one symbol time, amounting to 992 samples at the nominal 16 MHz mean PRF or 1016 samples at the nominal 64 MHz mean PRF. Figure 3.12 illustrates an example of the CIR, showing the normalized magnitudes. In this example, the CIR exhibits a rich multipath environment with multiple reflections and a dominant line-of-sight path. τ_0 represents the direct path, while τ_1 , τ_2 , and others correspond to the first, second, and subsequent reflections, respectively.

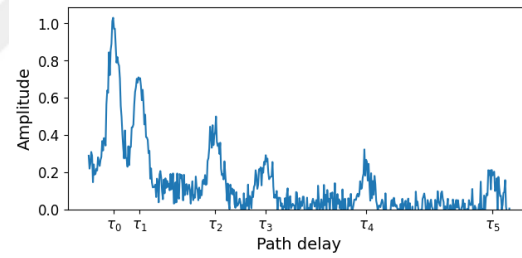


Figure 3.12. Normalized CIR magnitudes in a multipath scenario.

Following features were used in the model:

- (i) The measured distance from the UWB system. Measured distance is the primary feature and is the most correlated with ranging bias since DW1000 transceivers can not accurately estimate the received signal level (RSL). Figure 3.13 shows the measured distance from the UWB system. This makes distance a better indicator of RSL than the approximated RSL itself.
- (ii) Received power level, denoted as RX power, is an important feature for assessing the quality of the received signal in UWB ranging systems. It provides an estimate of the power level of the received signal, which can help determine the

accuracy of distance measurements. The DW1000 transceiver enables the estimation of received signal power using the Channel Impulse Response (CIR) power, as described in its user manual. The received power level (in dBm) can be estimated using the following formula:

$$RX \text{ Level} = 10 \times \log_{10} \left(\frac{C \times 2^{17}}{N^2} \right) - A, \quad (3.3)$$

where C is the Channel Impulse Response Power value reported in the CIR_PWR field of the Rx Frame Quality Information register, A is a constant (113.77 for a Pulse Repetition Frequency (PRF) of 16 MHz, or 121.74 for a PRF of 64 MHz), and N is the Preamble Accumulation Count value reported in the RXPACC field of the RX Frame Information register, which is also used as a standalone feature in the model.

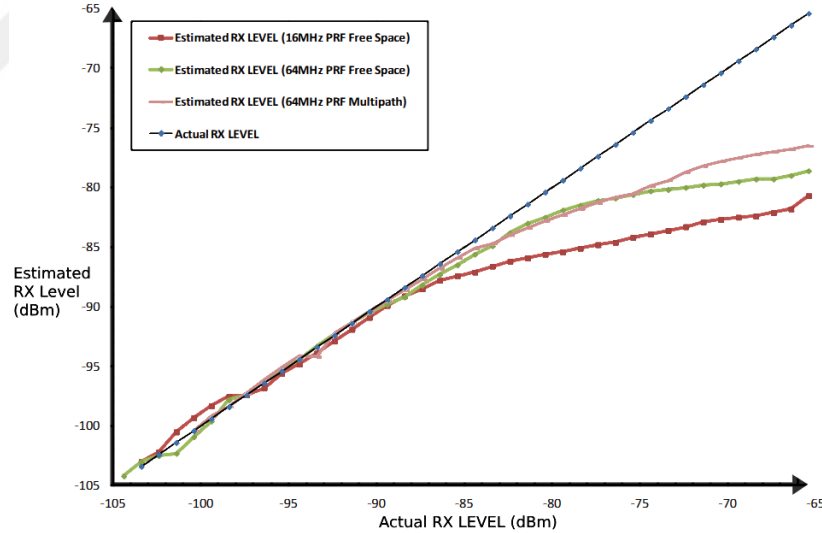


Figure 3.13. RSL estimation compared to actual RSL using DW1000 transceiver.

This estimation is generally accurate at lower received power levels but tends to underestimate the actual power at higher levels.

- (iii) The first path power level is a key parameter for understanding the direct signal reception in UWB systems. It provides an estimate of the power associated with the first detected signal path, which is critical for accurate ranging and determining the presence of multipath interference. The first path power level

(in dBm) is calculated using the following formula:

$$\text{First Path Power Level} = 10 \times \log_{10} \left(\frac{F_1^2 + F_2^2 + F_3^2}{N^2} \right) - A, \quad (3.4)$$

where F_1 is the First Path Amplitude (point 1) magnitude value reported in the FP_AMPL1 field of the Receive Time Stamp register, F_2 is the First Path Amplitude (point 2) magnitude value reported in the FP_AMPL2 field of the Rx Frame Quality Information register, F_3 is the First Path Amplitude (point 3) magnitude value reported in the FP_AMPL3 field of the Rx Frame Quality Information register, A is a constant (113.77 for a Pulse Repetition Frequency (PRF) of 16 MHz, or 121.74 for a PRF of 64 MHz), and N is the Preamble Accumulation Count value reported in the RXPACC field of the RX Frame Information register. The parameters F_1 , F_2 , and F_3 are also used as standalone features in the model.

- (iv) Leading Edge Detection (LDE) is a crucial process in the DW1000 transceiver for accurately identifying the first path of a received signal. It helps in distinguishing the direct path from multipath reflections, which is essential for precise ranging measurements. One of the key parameters derived from LDE is the maximum noise level, referred to as Max Noise in the DW1000 API. This value corresponds to the LDE_THRESH register (0x2E) and is determined based on an estimate of the noise obtained during the LDE algorithm's analysis of the accumulator data. A higher value indicates a noisier environment, which may affect the reliability of distance estimations.
- (v) First Path Amplitude 1 (FP_AMPL1) represents the magnitude of the leading edge signal detected by the DW1000 transceiver. It is a 16-bit unsigned value stored in the FP_AMPL1 field of register 0x15 – Receive Time Stamp.
- (vi) First Path Amplitude 2 (FP_AMPL2) is another 16-bit value that helps in determining the quality of the received signal. It is stored in register 0x12 – Rx Frame Quality Information.
- (vii) First Path Amplitude 3 (FP_AMPL3) is the last of the three amplitude values used for signal assessment. Like FP_AMPL2, it is a 16-bit value stored in register 0x12.

- (viii) Standard Noise (STD_NOISE) is a measure of the standard deviation of the noise level observed during the Leading Edge Detection (LDE) algorithm's analysis of the accumulator data.
- (ix) Channel Impulse Response Power (CIR_PWR) is a measure of the power received from the strongest portion of the signal channel.
- (x) RX Preamble Count (RXPACC) represents the number of accumulated preamble symbols successfully received by the DW1000 transceiver.
- (xi) First Path Index (FP_INDEX) is a 16-bit value stored in register 0x15 – Receive Time Stamp. It provides a CIR accumulator index value for the first path detected by the DW1000 transceiver. It is a 16-bit floating-point value where most significant 10 bits represent the integer part, and the least significant 6 bits represent the fractional part.
- (xii) Peak Path Index (LDE_PPINDEX) is a 16-bit status value stored in sub-register 0x2E:1000 of the Leading Edge Detection Interface. It provides a CIR accumulator index value for the peak path detected by the DW1000 transceiver.
- (xiii) Peak Amplitude (LDE_PPAMPL) is a 16-bit status value stored in sub-register 0x2E:1002 of the Leading Edge Detection Interface. It represents the amplitude of the peak path detected by the DW1000 transceiver.

To train the MLP model, the tag and two anchors (one being the customized anchor) were positioned facing each other, as illustrated in Figure 3.14. The tag was connected to a PC via USB, allowing data to be collected. It was moved back and forth few times towards the anchors to collect distance measurements, which were used to train the model. After the reception of the each final message in the distance measurements, all discussed features used in this work were extracted from the DW1000 transceiver. Several key parameters, including max noise, first path amplitude 1, first path amplitude 2, first path amplitude 3, standard noise, channel impulse response power, RX preamble count, and first path index, were obtained using the DW1000 API function `dwt_readdiagnostics`. Additionally, the peak path index and peak amplitude were acquired manually using the `dwt_read16bitoffsetreg` function. Since received power level and first path power require additional computations, they were

derived after obtaining the necessary raw values from the transceiver. During the setup, 51226 distance measurements were recorded alongside used features and laser sensor measurements for ground truth.

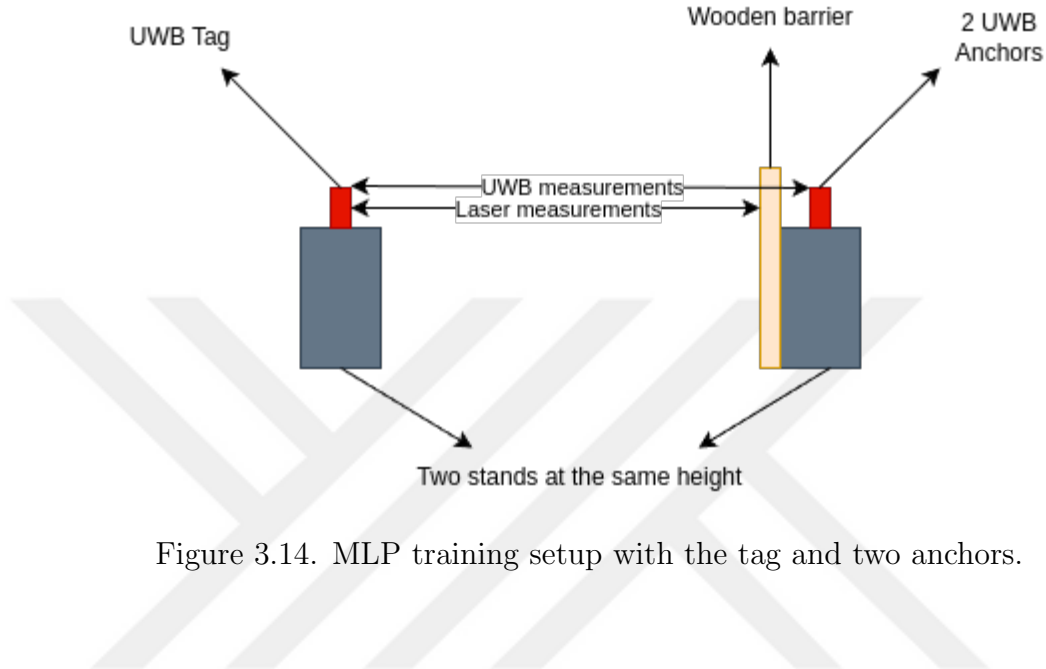


Figure 3.14. MLP training setup with the tag and two anchors.

Figure 3.15 presents the measurements reported by two different anchors. The left side of the figure shows laser-based measurements, while the right side displays uncalibrated UWB measurements. Although the UWB measurements appear slightly noisier, they exhibit a strong correlation with the laser measurements. However, the effect of ranging bias cannot be clearly observed, as the large number of samples makes it difficult to distinguish individual deviations within the figure.

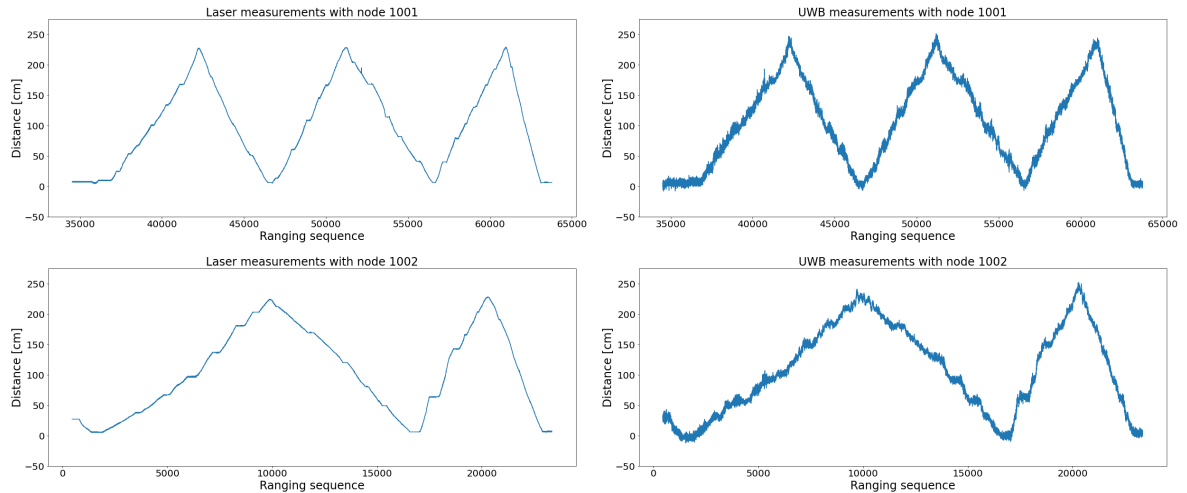


Figure 3.15. Laser and UWB distance measurements during data collection.

Figure 3.16 provides a closer view of the measurements, highlighting the differences between the laser and UWB distances. Red solid line represents raw UWB measurements while the blue solid line represents laser measurements. The UWB measurements exhibit a varying offset from the laser measurements, indicating the presence of ranging bias. This bias-induced error can lead to significant inaccuracies in the system, underscoring the importance of calibration. In this work, the MLP model was trained using the collected data to correct these errors and enhance the accuracy of UWB measurements.

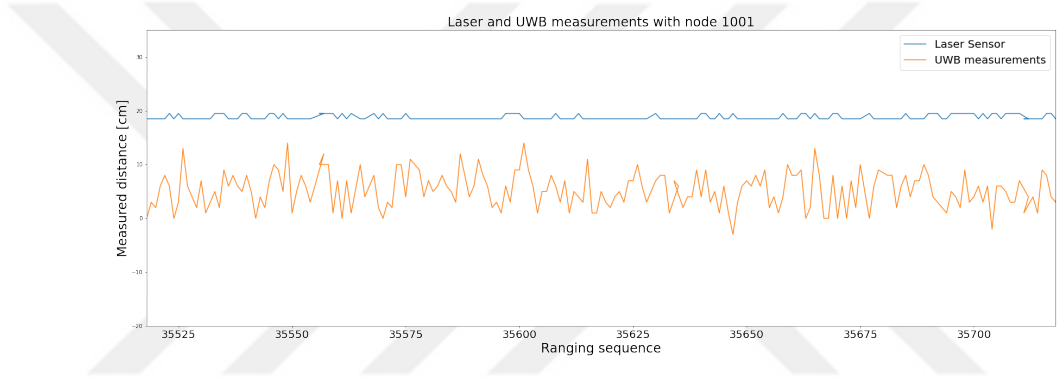


Figure 3.16. Close-up view of laser and UWB distance measurements.

Figure 3.17 illustrates the architecture of the MLP model used in this work. The model consists of an input layer with 13 features, followed by two hidden layers with 9 and 4 neurons, respectively, and a single neuron in the output layer. After the training data was collected, the model was trained using the Python `scikit-learn` library. The activation function used was ReLU, and all input features and the output were scaled using a standard scaler.

The choice of using fewer layers helps to reduce computational cost and lowers the likelihood of overfitting, making the model more efficient and generalizable for real-time applications. By incorporating multiple diagnostic features and leveraging the predictive capabilities of the MLP, this approach provides a dynamic and accurate method for correcting UWB ranging errors. Compared to traditional lookup table methods, the machine learning model achieves improved accuracy and adaptability in

diverse operational conditions.

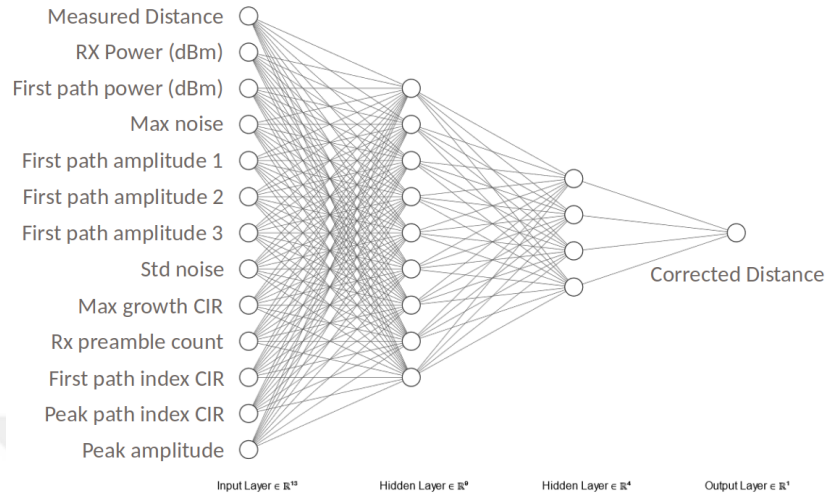


Figure 3.17. MLP layers.

Without any calibration, the root mean square error (RMSE) of the UWB system was measured to be 9.26 cm. After applying the MLP model, the RMSE was significantly reduced to 4.35 cm, indicating a calibration improvement of approximately 53%. This reduction demonstrates the effectiveness of the machine learning-based correction approach in enhancing the accuracy of UWB measurements.

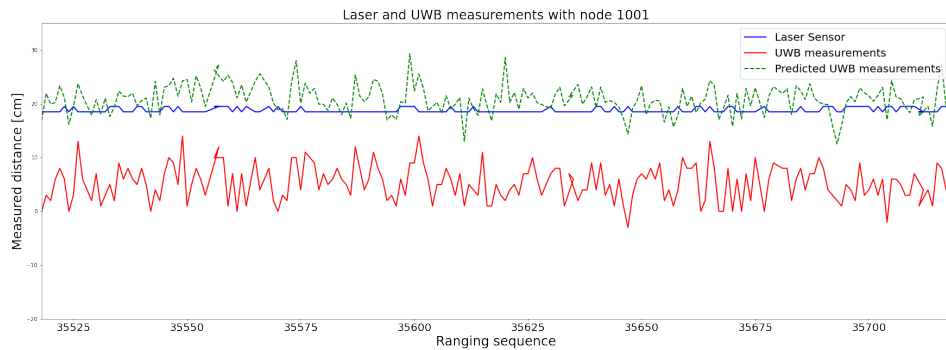


Figure 3.18. Close-up view of laser and UWB distance measurements after MLP correction.

Figure 3.18 shows the laser and UWB distance measurements after applying the MLP correction. Dotted green line represents corrected UWB measurements. The

UWB measurements now exhibit a much closer alignment with the laser measurements, indicating a significant reduction in ranging bias. The improved accuracy achieved through the MLP model calibration underscores the effectiveness of machine learning in enhancing UWB ranging precision.

To enable real-time correction of UWB ranging errors in an embedded system, the trained MLP model needed to be converted into an efficient C implementation. Running the model in Python is not suitable for embedded applications due to computational and memory constraints. Therefore, a tool was required to transpile the trained model into optimized C code while preserving its accuracy and performance.

CLARA is an open-source tool specifically designed to facilitate this process by converting machine learning models into C for seamless deployment in embedded systems [50]. Unlike traditional code conversion tools, CLARA does not simply translate Python scripts into C but instead transpiles the trained model itself into efficient, hardware-optimized C code. This approach allows for real-time execution with minimal computational overhead, making it well-suited for embedded applications. In this work, CLARA was used to convert the trained MLP model into C source code, ensuring high-performance inference on the target hardware.

3.4. Real-time Implementation

3.4.1. Initialization and Configuration of the UWB Transceiver

The UWB system was initialized and configured using the DW1000 API to ensure that both tag and anchor nodes are ready for ranging measurements. The process began by resetting the DW1000 via its RESET pin, which is connected to a microcontroller GPIO configured in open-drain mode; the pin is pulled low for a few milliseconds before being set to high-impedance, thereby allowing the DW1000 to resume normal operation. Next, the `dwt_initialize` function was called to establish communication and verify the device ID (expected value `0xDECA0130`). After successful initialization,

the transceiver was configured using `dwt_configure` with parameters set to Channel 5, a data rate of 6.8 Mbps, a preamble length of 128 symbols, and a PRF of 64 MHz.

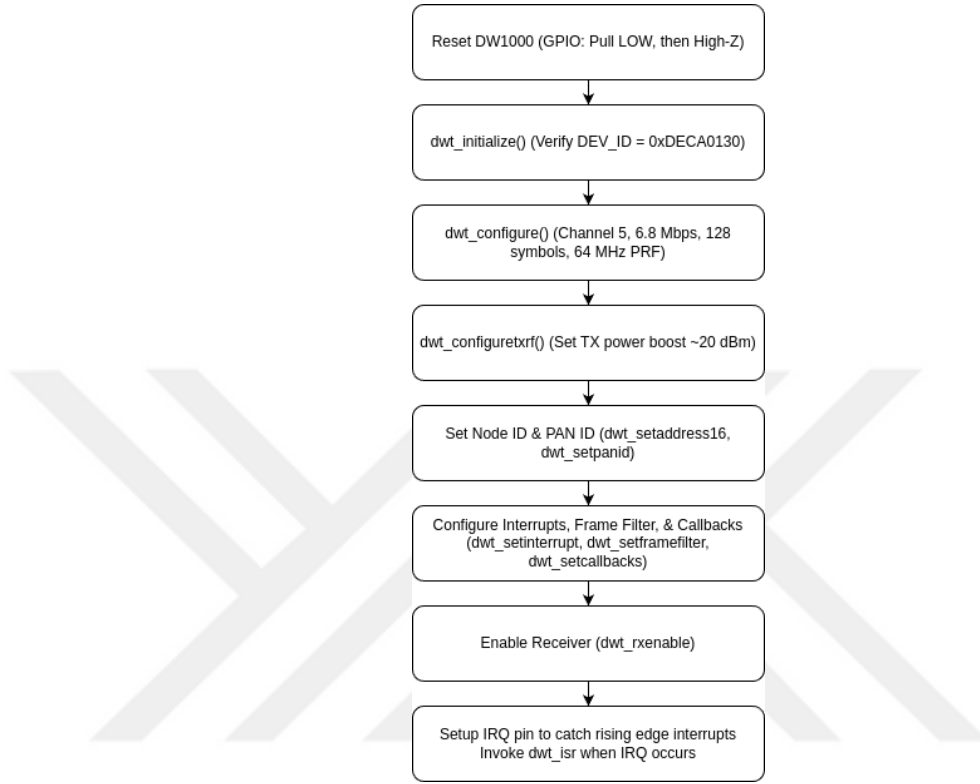


Figure 3.19. DW1000 initialization flow diagram.

Table 3.1. UWB configuration parameters used in this work.

Parameter	Value
Channel	5
Preamble length	128
Preamble code	9
Bit rate	6.8 Mbps
PRF	64 MHz

Transmitter parameters were then set via `dwt_configurertxrf` (applying an internal PA boost of approximately 20 dBm), followed by setting the node and PAN IDs using `dwt_setaddress16` and `dwt_setpanid`. Finally, interrupt generation, frame filtering, and callback functions were configured using `dwt_setinterrupt`, `dwt_setframe-`

`filter`, and `dwt_setcallbacks`; for anchor nodes, the receiver was enabled continuously with `dwt_rxenable`. The GPIO pin associated with the DW1000's IRQ line was configured as an input with a rising edge interrupt to detect DW1000 events. When an interrupt occurs, the handler calls the `dwt_isr` function, which reads the DW1000 status register to identify the specific event. Based on the event detected, the corresponding callback function is invoked, if one has been defined. The initialization flow diagram is shown in Figure 3.19.

3.4.2. Sending and Receiving UWB Messages

After the initialization of the UWB transceiver, the system was ready to send and receive messages for ranging. To receive a frame, when the device is in IDLE mode, the receiver is enabled using `dwt_rxenable` with the `DWT_START_RX_IMMEDIATE` flag. The device then waits for a frame; upon reception, the registered callback function for the RXFCG (Receive Frame Control Good) event is invoked. The received frame is subsequently read using `dwt_readrxdata` and processed, while its corresponding timestamp can be obtained via `dwt_getrxtimestamp`.

To send a frame, the transmitter is first forced into IDLE mode using `dwt_forcetrxoff`. The MCU then prepares the frame according to the IEEE 802.15.4a standard—which permits MAC frames up to 127 bytes, including the MAC header and frame check sequence (FCS) fields. In this work, all frames were sent using `0x4188` as the frame control field, indicating a data frame with 16-bit addressing, no acknowledgment, and no security. The prepared frame is written to the transceiver using `dwt_writetxdata` and `dwt_writetxfctrl`, and transmission is initiated with `dwt_starttx`.

For delayed transmission, `dwt_setdelayedtrxtime` is used to specify the desired transmission time. The DW1000's clock is based on a 40-bit counter with a resolution of 15.65 ps; however, in delayed transmission the lower 9 bits are not used, so the high 31 bits are supplied, yielding an effective timing resolution of approximately 8 ns. In

such cases, `dwt_starttx` must be called with the `DWT_START_TX_DELAYED` flag.

3.4.3. Modified Two-Way Ranging Protocol

To achieve a high update rate of 100 Hz, modifications were made to the standard TWR protocol. Instead of performing TWR measurements sequentially with each anchor, the protocol was adjusted so that all four anchors respond within a single TWR sequence. This modification significantly reduces the total measurement duration while maintaining accuracy.

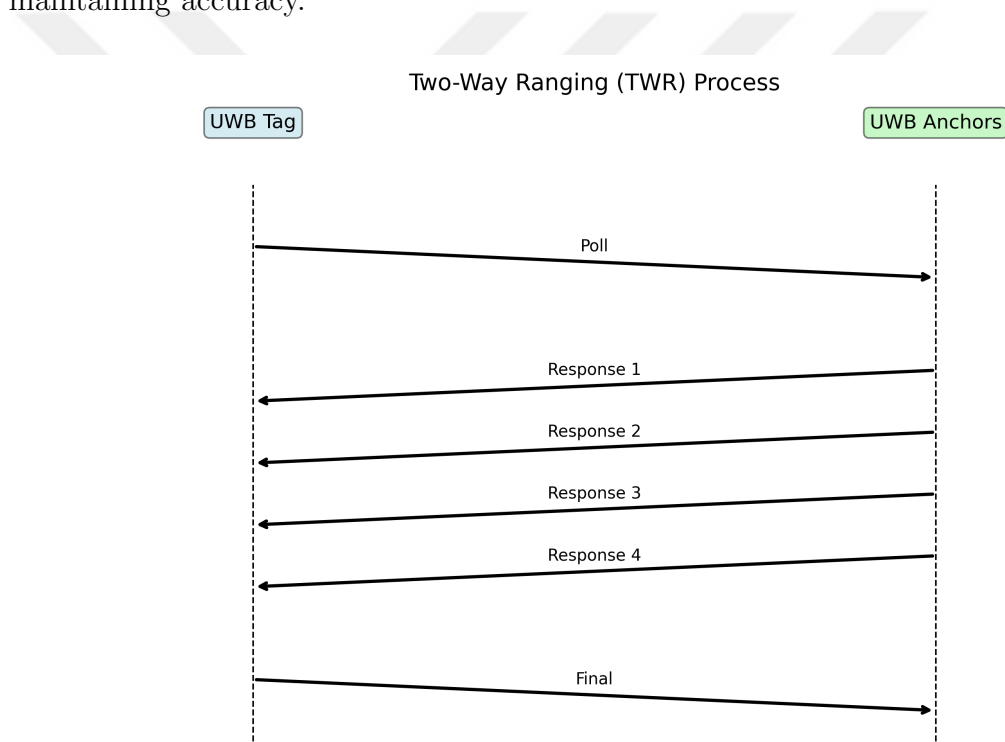


Figure 3.20. TWR process with multiple anchors.

In the modified protocol, the delay between the *poll* message and the first *response* message is set to $1000\ \mu s$, while the delay between consecutive *response* messages from the anchors is $600\ \mu s$. Additionally, a delay of $1000\ \mu s$ is maintained between the final *response* message and the subsequent *final* message. These adjustments yield a complete TWR sequence duration of approximately $3.8\ ms$ when all four anchors are involved.

The tag initiates the TWR sequence by broadcasting a *poll* message to all anchors. Following this transmission, the tag waits for a turnaround period of approximately $1000\ \mu s$ before switching to receive (RX) mode, in which it anticipates the incoming *response* messages from the anchors. Each *response* message, upon reception, triggers a separate RXFCG (Receive Frame Control Good) event; however, because the transceiver reverts to IDLE mode following the event, the RX mode must be reenabled at the conclusion of each callback to ensure continuous readiness for subsequent responses. The tag buffers each received *response* message along with its corresponding timestamp in the MCU's RAM.

Tag TWR algorithm:

Require: The UWB transceiver is initialized and configured.

Ensure: TWR sequence with poll and final messages transmitted, and responses received with timestamps recorded.

- 1: **Broadcast** the *poll* message to all anchors.
- 2: **Wait** for a turnaround time of $1000\ \mu s$.
- 3: **Switch** to RX mode using `dwt_rxenable(DWT_START_RX_IMMEDIATE)`.
- 4: **while** the response reception window is active **do**
- 5: **if** a *response* is received (RXFCG event) **then**
- 6: **Read** the response message and record its timestamp using `dwt_readrxdata` and `dwt_getrxtimestamp`.
- 7: **Re-enable** RX mode.
- 8: **end if**
- 9: **end while**
- 10: **Prepare** the *final* message, ensuring that all timestamps (including the pre-calculated final transaction timestamp) are included.
- 11: **Broadcast** the *final* message (with destination address `0xFFFF`) to all anchors.

Figure 3.21. Tag TWR algorithm.

After a predetermined interval that allows all anchors to transmit their responses, the tag compiles and broadcasts a *final* message to all anchors. Notably, both the *poll*

and *final* messages are broadcast messages, with the destination address in the MAC frame set to $0xFFFF$.

Anchor TWR algorithm:

Require: The UWB transceiver is initialized and configured in continuous RX mode.

Ensure: A response message is transmitted and the final message is processed to compute distance.

- 1: Wait for a *poll* message from the tag.
- 2: Upon receiving a *poll* message, **prepare** the *response* message.
- 3: **Calculate** the delayed transmission time:

$$T_{\text{delay}} = T_{\text{poll_rx}} + 1000 \mu s + 600 \mu s \times (\text{anchor_id mod } 10)$$

- 4: **Send** the *response* message using delayed transmission.
- 5: Wait for the *final* message from the tag.
- 6: Upon receiving the *final* message, extract the relevant timestamps.
- 7: **Compute** the distance using the collected timestamps.
- 8: Optionally, **transmit** the computed distance to the tag.

Figure 3.22. Anchor TWR algorithm.

Anchors, in contrast, remain in RX mode almost continuously. Upon receiving a *poll* message from the tag, an anchor immediately triggers an RX callback and prepares a *response* message, which is then transmitted using delayed transmission. The delayed transmission interval is computed as $1000 + 600 \times (\text{anchor_id mod } 10) \mu s$. It means that the first anchor to respond has a delay of $1000 \mu s$, the second anchor has a delay of $1600 \mu s$, the third anchor has a delay of $2200 \mu s$, and the fourth anchor has a delay of $2800 \mu s$. This staggered transmission ensures that the *response* messages do not collide. After transmitting its *response* message, an anchor continues to monitor for the *final* message from the tag. Once the *final* message is received, the anchor is able to compute the distance by processing all the relevant timestamps, and it subsequently communicates this distance to the tag via a response message.

This approach significantly increases the update rate of the UWB-based localization system while maintaining compatibility with existing TWR protocols. The modifications draw inspiration from the N-TWR protocol described in [51], which introduces efficient methods for handling multiple responders in a single ranging sequence.

In this work, the payload of the *poll*, *response*, and *final* messages was fully customized to include additional information required for real-time localization and orientation tracking:

- The *poll* message payload includes the laser sensor value, the most recent values of orientation quaternions, and acceleration data in the tag's body frame.
- The *response* message from each anchor contains the last calculated (x, y) position of the tag. Additionally, the response from the customized anchor includes the anchor's quaternions, which represent the base's orientation in the 3D space.

This customized payload design allows the tag to be aware of its own location and the base orientation in near real-time, significantly enhancing the system's responsiveness and accuracy.

This approach significantly increases the update rate of the UWB-based localization system while maintaining compatibility with existing TWR protocols. The modifications draw inspiration from the N-TWR protocol described in [51], which introduces efficient methods for handling multiple responders in a single ranging sequence.

3.4.4. Data Flow and Communication

In the proposed system, each UWB anchor is connected to the Raspberry Pi 5 via USB. The DWM1001-DEV module, while equipped with a USB interface, internally uses a UART-to-USB bridge. On the microcontroller side, communication is handled through UART with the following configuration: a baud rate of 230400, 8 data bits, no parity, and no flow control. This setup ensures fast and reliable data transfer between

the anchors and the Raspberry Pi.

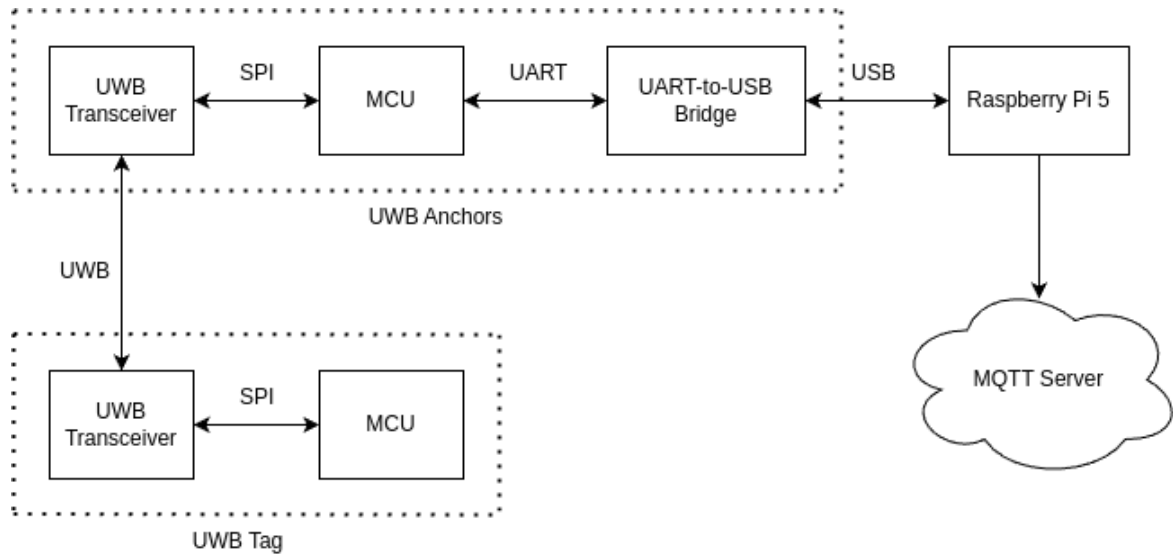


Figure 3.23. Data acquisition system diagram.

At the end of each TWR session, each anchor calculates the Time-of-Flight (ToF) and converts it into a distance measurement in centimeters. In addition to the distance, the anchors also collect the RX diagnostics values extracted from final message of the TWR session which are used as features for the MLP model, along with the tag's acceleration and orientation represented as quaternions. This data is then packed into a TinyFrame message and sent over the UART interface. TinyFrame is a lightweight library designed for building and parsing data frames to be transmitted over serial interfaces such as UART, telnet, or sockets. Written in C, the library is compatible with `--std=gnu99` and largely supports `--std=gnu89`, making it versatile for embedded and general-purpose applications. TinyFrame provides a high-level interface for managing communication sessions between two peers. Features such as multi-message sessions, response listeners, checksums, and timeouts are built into the library, simplifying the implementation of reliable communication protocols [52].

The Raspberry Pi hosts an MQTT server, which facilitates the communication of calculated locations to external systems. After processing the data and estimating the tag's position, the Raspberry Pi publishes the calculated X and Y coordinates on

the topic "location/estimated". The published data is formatted in JSON, allowing easy integration with other systems and applications.

3.4.5. Location Estimation

The EKF is used to estimate the UAV's state, including its position and velocity, by combining UWB distance measurements with dynamic acceleration data. As C implementation a library called **TinyEKF** was used [53]. TinyEKF is lightweight and designed for embedded systems, making it suitable for real-time applications. This subsection details the process, including dynamic acceleration filtering, quaternion-based rotation, and the matrices used in the EKF.

The overall system architecture for real-time location estimation is depicted in Figure 3.24. For each UWB ranging sequence, the acceleration values are first processed by removing the gravitational component and then rotated into the localization reference frame. The resulting vector is used as the input for the EKF predict step.

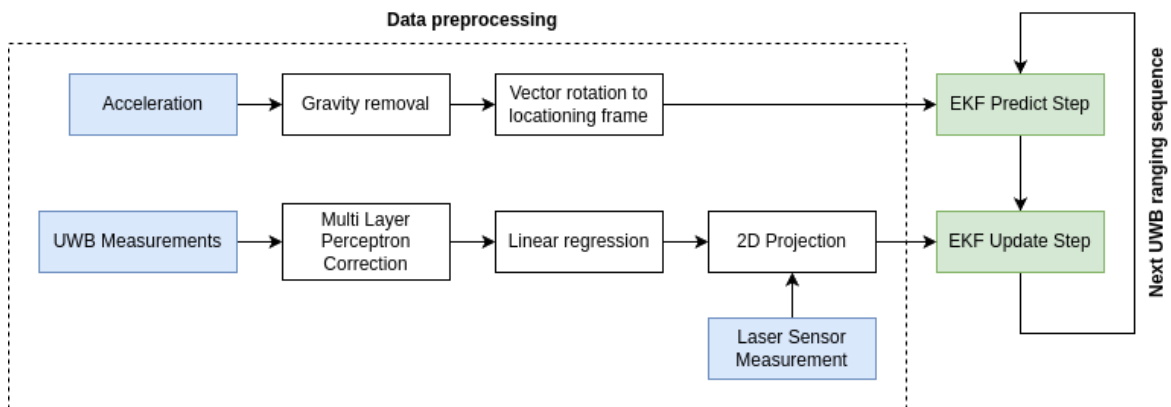


Figure 3.24. Locationing flowchart.

For the UWB measurements, each one is initially corrected using a Multi-Layer Perceptron (MLP) model. Following this, linear regression is applied to filter out outliers and predict measurements in cases where data is missing for a specific anchor. The corrected measurements are then projected into the 2D plane using the laser sensor for altitude information. These resulting measurements serve as the input for the EKF

update step.

The EKF predict and update steps are executed recursively for each new UWB measurement sequence, ensuring robust and accurate real-time localization.

3.4.5.1. Dynamic Acceleration Filtering. To extract dynamic acceleration $\mathbf{a}_{\text{dynamic}}$, the static component of acceleration due to gravity $\mathbf{g}$ is removed. Gravity is estimated using a first-order Infinite Impulse Response (IIR) filter:

$$\mathbf{g}(t) = \alpha \mathbf{g}(t-1) + (1 - \alpha) \mathbf{a}(t), \quad (3.5)$$

where:

- $\alpha = \frac{\tau}{\tau + \Delta t}$ is the smoothing coefficient, with τ as the filter time constant and Δt as the sampling interval.
- $\mathbf{a}(t)$: Raw acceleration vector at time t .
- $\mathbf{g}(t)$: Estimated gravity vector.

The dynamic acceleration is then:

$$\mathbf{a}_{\text{dynamic}}(t) = \mathbf{a}(t) - \mathbf{g}(t). \quad (3.6)$$

3.4.5.2. Acceleration Vector Rotation. The IMU's acceleration readings are in the tag's body frame, which must be rotated to the global ENU frame. This is achieved using quaternions. Let:

- $\mathbf{q}_{\text{tag}}$: Quaternion representing the tag's orientation in the global frame.
- $\mathbf{q}_{\text{base}}$: Quaternion representing the base anchor's orientation in the global frame.
- $\mathbf{a}_{\text{dynamic}}$: Dynamic acceleration in the tag's local frame.
- $\mathbf{a}_{\text{ENU}}$: Dynamic acceleration in the ENU frame.

The rotation is performed as

$$\mathbf{a}_{\text{ENU}} = \mathbf{q}_{\text{base}}^* \otimes (\mathbf{q}_{\text{tag}} \otimes \mathbf{a}_{\text{dynamic}} \otimes \mathbf{q}_{\text{tag}}^*) \otimes \mathbf{q}_{\text{base}}. \quad (3.7)$$

This aligns the acceleration data with the ENU frame, ensuring accurate motion modeling in the prediction step of the EKF.

3.4.5.3. Extended Kalman Filter Matrices. The EKF state vector x contains the UAV's position and velocity as follows:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ v_x \\ v_y \end{bmatrix}, \quad (3.8)$$

where x_1, x_2 represent the position in the ENU frame, and v_x, v_y represent the velocity in the ENU frame.

The state transition model is

$$f(x, u) = \begin{bmatrix} x_1 + v_x \Delta t \\ x_2 + v_y \Delta t \\ v_x + a_{\text{ENU},x} \Delta t \\ v_y + a_{\text{ENU},y} \Delta t \end{bmatrix}. \quad (3.9)$$

Here, $u = [a_{\text{ENU},x}, a_{\text{ENU},y}]^T$ is the control input derived from dynamic acceleration. The model assumes constant acceleration during the time step Δt , which simplifies the computation while maintaining reasonable accuracy for short intervals. This formulation is particularly useful in scenarios where the system dynamics are influenced by external forces, such as in UAV navigation or robotic motion planning. By incorporating the control input u , the model enables precise trajectory prediction and state estimation.

The Jacobian of the state transition function, F , is

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.10)$$

The control matrix B is

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}. \quad (3.11)$$

The process noise covariance matrix Q , accounting for acceleration noise σ_a^2 , is

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix}. \quad (3.12)$$

The measurement function $h(x)$ maps the state to the UWB distance measurements

$$h(x) = \begin{bmatrix} \sqrt{(x_1 - a_{1x})^2 + (x_2 - a_{1y})^2} \\ \sqrt{(x_1 - a_{2x})^2 + (x_2 - a_{2y})^2} \\ \sqrt{(x_1 - a_{3x})^2 + (x_2 - a_{3y})^2} \\ \sqrt{(x_1 - a_{4x})^2 + (x_2 - a_{4y})^2} \end{bmatrix}, \quad (3.13)$$

where (a_{ix}, a_{iy}) are the anchor positions.

The Jacobian of the measurement function, H , is

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{x_1 - a_{1x}}{r_1} & \frac{x_2 - a_{1y}}{r_1} & 0 & 0 \\ \frac{x_1 - a_{2x}}{r_2} & \frac{x_2 - a_{2y}}{r_2} & 0 & 0 \\ \frac{x_1 - a_{3x}}{r_3} & \frac{x_2 - a_{3y}}{r_3} & 0 & 0 \\ \frac{x_1 - a_{4x}}{r_4} & \frac{x_2 - a_{4y}}{r_4} & 0 & 0 \end{bmatrix}, \quad (3.14)$$

where $r_i = \sqrt{(x_1 - a_{ix})^2 + (x_2 - a_{iy})^2}$.

The measurement noise covariance R is

$$R = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2), \quad (3.15)$$

where σ_i^2 represents the variance of the UWB distance measurements.

By applying the above matrices, the EKF predicts the state using motion data and updates it using UWB measurements to provide accurate real-time localization.

4. SYSTEM EVALUATION

4.1. Experimental Setup

Figure 4.1 illustrates the experimental setup used for evaluating the proposed UAV localization system. The setup consists of a rectangular foam board with dimensions $0.6 \times 0.7\text{m}$, serving as the base for mounting the UWB anchors. The four UWB anchors are placed at the corners of the foam board and are connected to a Raspberry Pi 5 via USB cables for data acquisition and processing. The Raspberry Pi is located at the center of the base and acts as the central computing unit for the system.

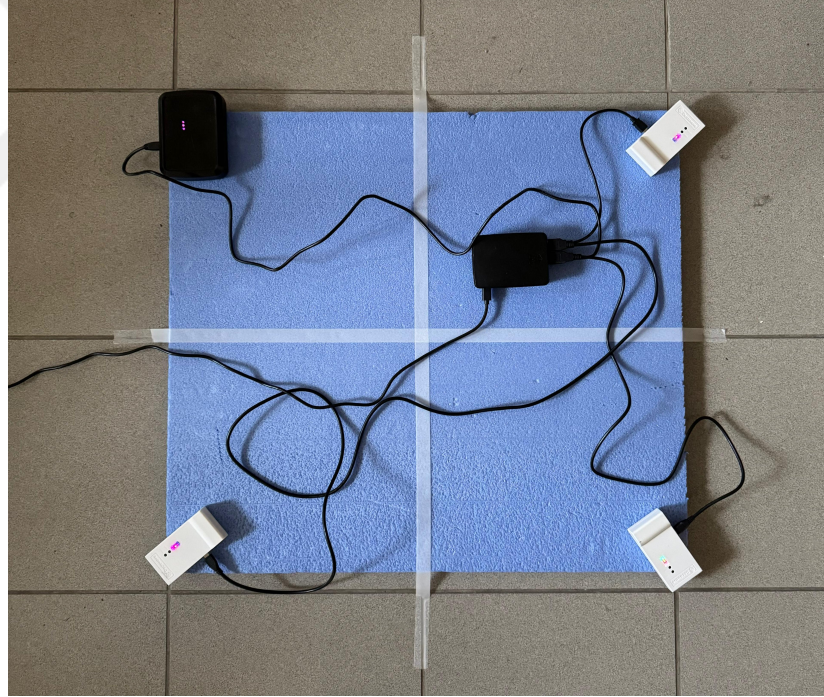


Figure 4.1. Experimental setup featuring UWB anchors.

The customized UWB anchor, positioned at the top-left corner of the foam board, is carefully aligned with the locationing reference frame. This alignment ensures that the anchor's orientation matches the East-North-Up (ENU) coordinate system used throughout the localization process. This alignment is crucial for accurate determination of the UAV's position and orientation.

4.2. Indoor Static Localization Test Results

For evaluating static localization accuracy, the UWB tag was mounted on a carton box with a height of 20 cm. The tag was then placed at predefined positions on the base board, with measurements taken using a ruler referencing the center of the board as the origin. The tag was positioned at the following coordinates (in meters): $(0, 1)$, $(0, 0)$, $(0, -1)$, $(1, 1)$, $(1, 0)$, $(1, -1)$, $(-1, 1)$, $(-1, 0)$, $(-1, -1)$, $(0.66, 0.66)$, $(-0.66, 0.66)$, $(0.66, -0.66)$, $(-0.66, -0.66)$.

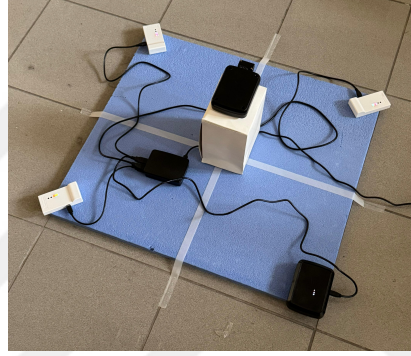


Figure 4.2. Static localization test with UWB tag placed at $(0, 0)$.

The positions of the UWB tag for two of these tests are shown in Figures 4.2 and 4.3. In Figure 4.2, the tag is positioned at the origin $(0, 0)$, while in Figure 4.3, it is located at $(-0.66, -0.66)$. All measurements were conducted without the aid of a ground truth localization system, relying on manual measurements for accuracy evaluation.

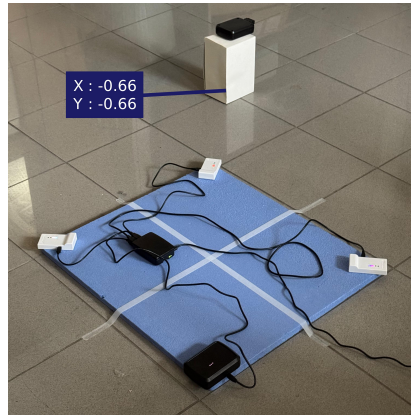


Figure 4.3. Static localization test with UWB tag placed at $(-0.66, -0.66)$.

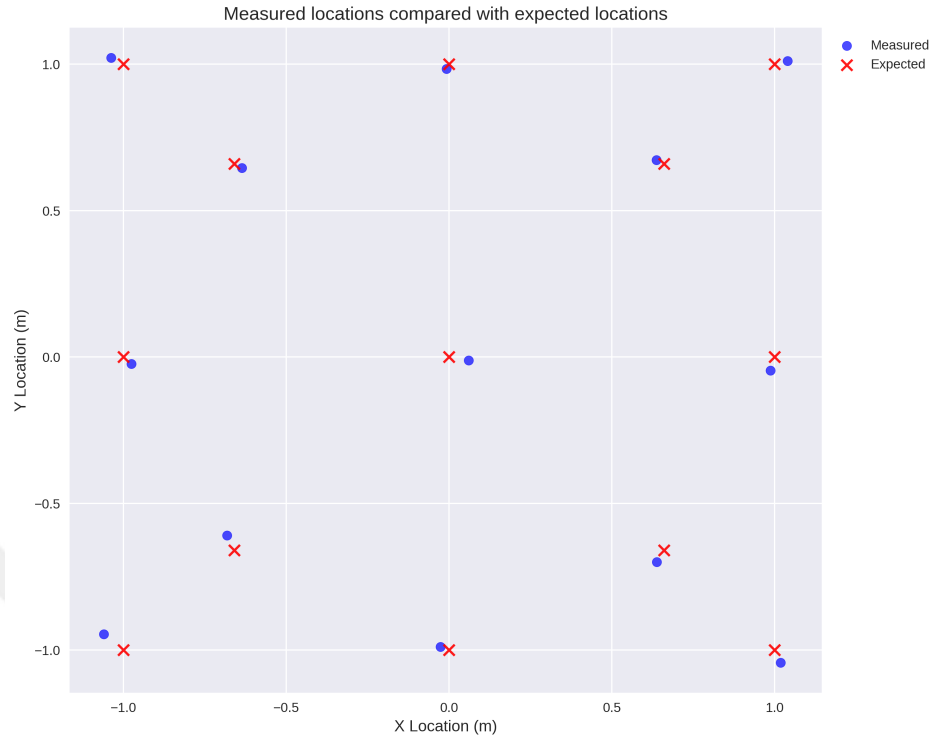


Figure 4.4. All measured and expected locations during static localization test.

Table 4.1. Static measurement results

Measured X (m)	Measured Y (m)	Expected X (m)	Expected Y (m)	RMSE (m)
-1.060545	-0.945927	-1.0	-1.0	0.081176
-0.975987	-0.023367	-1.0	0.0	0.033506
-1.03768	1.021492	-1.0	1.0	0.043378
-0.026959	-0.989685	0.0	-1.0	0.028866
0.06006	-0.011071	0.0	0.0	0.061071
-0.007603	0.983204	0.0	1.0	0.018437
1.017836	-1.043238	1.0	-1.0	0.046772
0.987003	-0.04584	1.0	0.0	0.047647
1.039657	1.010926	1.0	1.0	0.041135
-0.682285	-0.609458	-0.66	-0.66	0.055237
-0.636473	0.645925	-0.66	0.66	0.027416
0.638263	-0.699677	0.66	-0.66	0.045241
0.636618	0.672593	0.66	0.66	0.026557

Figure 4.4 shows the measured and expected locations of the UWB tag for all static localization tests. Table 4.1 presents the measured and expected locations of the UWB tag for the static localization tests, along with their respective root mean square error (RMSE) values. The lowest observed RMSE is approximately 1.8 cm, while the highest observed RMSE is 8.1 cm. The overall average RMSE across all test points is 4.28 cm. These results demonstrate that the system achieves high localization accuracy, which is critical for the precision requirements of an autonomous UAV landing system.

4.3. Indoor Dynamic Localization Test Results

Dynamic localization tests were conducted to evaluate the system's performance during movement. In the first test, the UWB tag was manually moved along a square path with a side length of 1 meter, anchored at the origin of the localization frame. Figure 4.5 illustrates the measured trajectory of the UWB tag during this test, demonstrating the system's ability to track movement along a predefined path accurately.

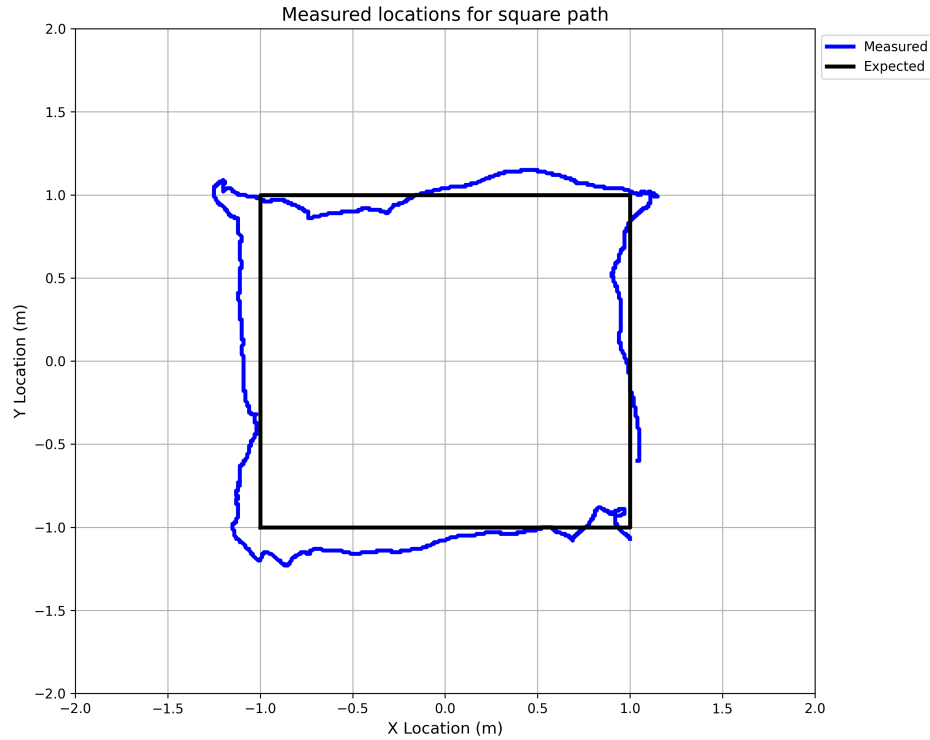


Figure 4.5. Measured trajectory of the UWB tag during the square path dynamic localization test.

In the second test, the UWB tag was manually moved to simulate a landing maneuver. The movement was guided by hand, following a trajectory resembling a UAV's descent. Figure 4.6 shows the measured trajectory during this test, highlighting the system's capability to track the dynamic motion of the tag accurately.

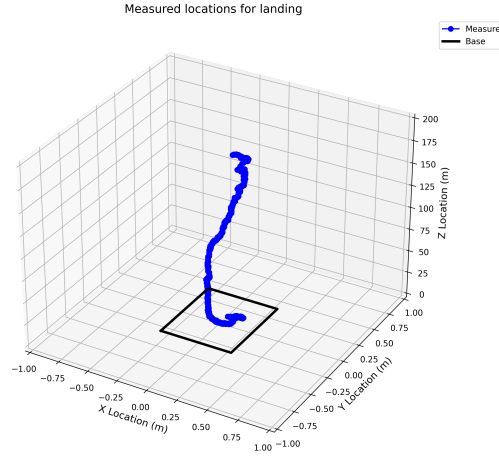


Figure 4.6. Measured trajectory of the UWB tag during the simulated landing trajectory test.

It is important to note that due to the lack of a ground truth system during these tests, precise metrics such as root mean square error (RMSE) or other performance parameters could not be calculated. Despite this limitation, the qualitative assessment of the measured trajectories suggests high accuracy and reliability, which are essential for autonomous landing scenarios for UAVs.

4.4. Outdoor Field Test Results

Field tests were conducted to evaluate and verify the operability and accuracy of the developed localization system in outdoor environments. The tests involved mounting an Ultra-Wideband (UWB) tag on a drone to observe and measure system performance, particularly during landing maneuvers.

A DJI Matrice 300 RTK drone was selected for this experimental study. The Matrice 300 RTK is a commercial-grade drone platform by DJI, characterized by ex-

tended flight capabilities, advanced AI functionalities, and robust sensing features. Key specifications are summarized below:

- **Dimensions:** Unfolded (excluding propellers): $810 \times 670 \times 430$ mm; Folded (including propellers): $430 \times 420 \times 430$ mm
- **Diagonal Wheelbase:** 895 mm
- **Weight:** Approximately 3.6 kg (without batteries); Approximately 6.3 kg (with dual TB60 batteries)
- **Maximum Payload:** 2.7 kg
- **Maximum Takeoff Weight:** 9 kg
- **Maximum Speed:** 23 m/s
- **Maximum Flight Time:** 55 minutes
- **Operating Temperature Range:** -20°C to 50°C
- **Ingress Protection Rating:** IP45
- **GNSS Support:** GPS, GLONASS, BeiDou, Galileo



Figure 4.7. DJI Matrice 300 RTK drone employed in field tests.

Due to inadequacies observed in preliminary tests using a lightweight foam platform, a more robust wooden platform measuring $0.9\text{ m} \times 1.1\text{ m}$ was constructed. This new platform provided sufficient stability and resistance to aerodynamic disturbances caused by the drone's propellers. For precise positioning and successful landing maneuvers, UWB anchors were installed at each corner of the platform, ensuring comprehensive coverage and clear line-of-sight communication with the drone-mounted UWB tag.

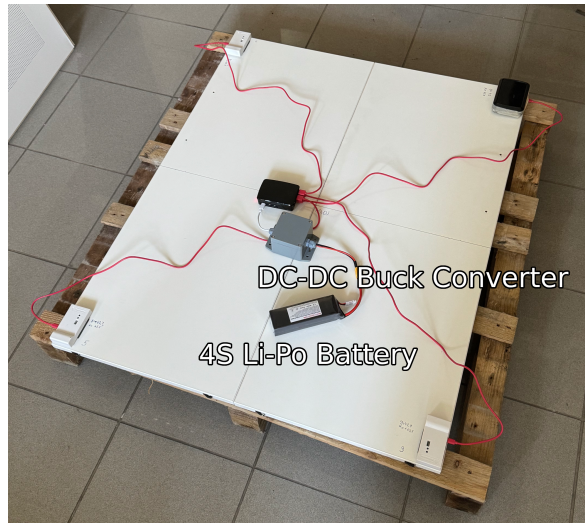


Figure 4.8. Configuration of the localization system during outdoor field tests.

Considering the outdoor testing scenario, a battery-based power solution replaced the AC adapter utilized in earlier indoor experiments. A 14.8 V (4S) lithium-polymer battery supplied power through a DC-DC buck converter, which was capable of delivering up to 100 W (20 A) while maintaining a regulated 5.1 V output. Prior validation using an electronic load confirmed stable voltage performance under varying load conditions. Subsequently, the power module was integrated into an enclosure along with necessary cabling and mounted beneath the wooden platform for protection against environmental factors and drone operations.

Since the Raspberry Pi 5, used as the processing unit, required USB Power Delivery (PD) compliance via its USB Type-C port, the device was configured to draw currents exceeding the default 600 mA limitation. This configuration involved adding the following parameter to the Raspberry Pi's `config.txt` file:

```
usb_max_current_enable=1
```

After the complete assembly of the power and localization modules, preliminary indoor validations reconfirmed system accuracy. Subsequently, outdoor tests were con-

ducted at the Uçaksavar Campus of Boğaziçi University, chosen for its open-field environment suitable for drone operations.

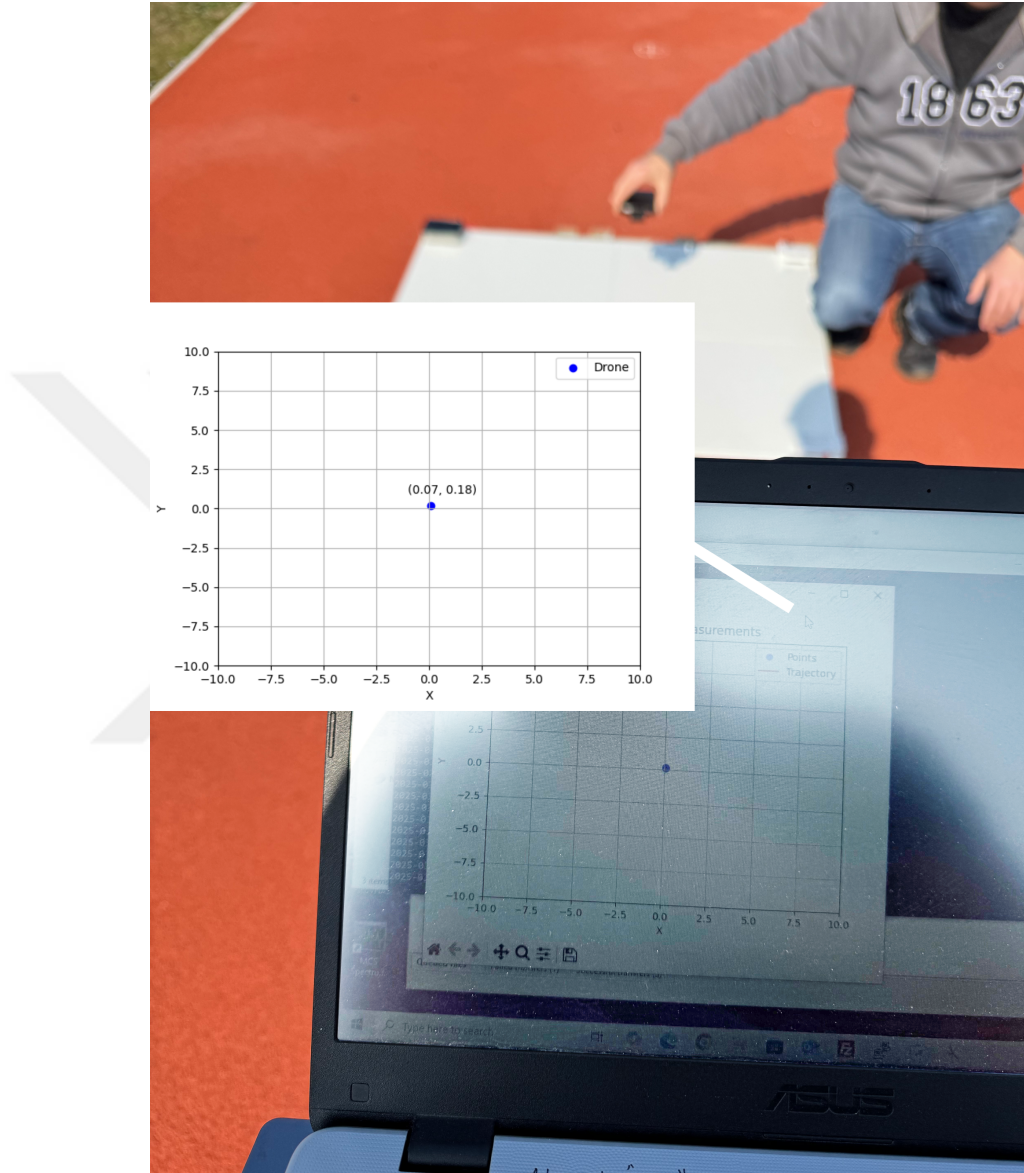


Figure 4.9. UWB tag manually placed at the origin (0 m, 0 m).

The first set of tests involved manually positioning the UWB tag at predefined coordinates, including the origin (0 m, 0 m), (2.5 m, 0 m), (5 m, 0 m), (7.5 m, 0 m), and (10 m, 0 m), relative to the platform's center.

As illustrated in Figure 4.9, the estimated coordinates closely matched the actual positions (estimated coordinates: X: 0.07 m, Y: 0.18 m, Z: 0.27 m).

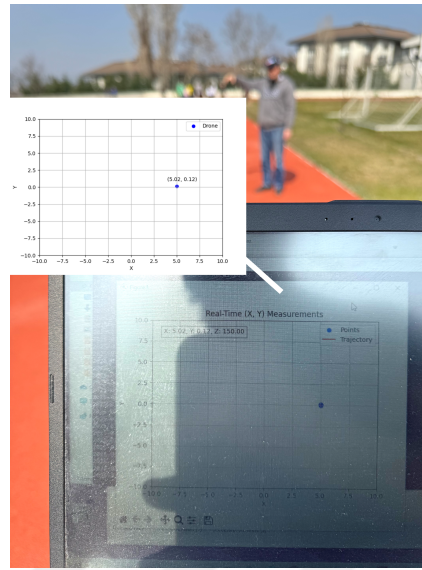


Figure 4.10. UWB tag manually placed at coordinates (5 m, 0 m).

Similar accuracy was achieved at (5 m, 0 m), where the estimated position was recorded as X: 5.02 m, Y: 0.12 m, Z: 1.5 m.

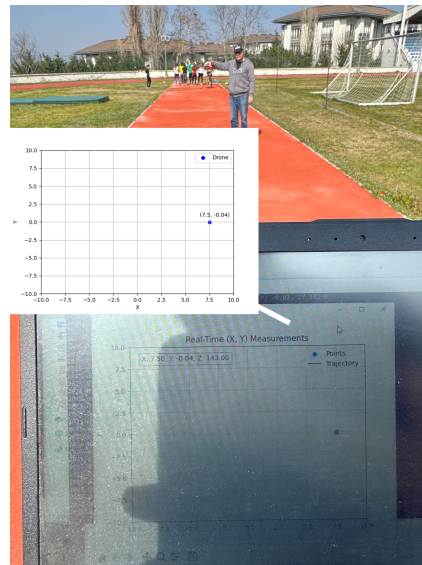


Figure 4.11. UWB tag manually placed at coordinates (7.5 m, 0 m).

The consistency and precision of the localization system were further confirmed at subsequent test points.



Figure 4.12. Drone during mid-flight maneuvers.

The final phase involved mounting the UWB tag on the drone itself, performing manual flight maneuvers, and simulating landing sequences. During drone flights conducted at approximately 1.5 meters above ground level, the real-time position estimates displayed on the user interface consistently matched expected flight paths and landing zones, confirming the system's suitability for precise autonomous drone landings.

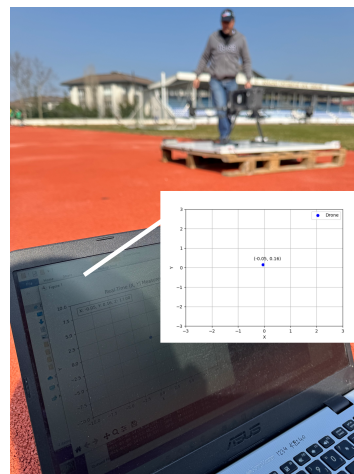


Figure 4.13. Drone at the landing platform center.

Overall, the outdoor field tests successfully demonstrated the robustness, relia-

bility, and accuracy of the developed UWB-based localization system under realistic operational conditions.



5. CONCLUSION

The development and evaluation of a robust localization system for UAV autonomous landing have been the primary focus of this thesis. By leveraging UWB technology, advanced machine learning techniques, and a tightly integrated sensor fusion framework using the EKF, this research successfully established an accurate solution for UAV localization and landing applications.

The system underwent extensive validation through both indoor and outdoor experiments. Indoor tests included static and dynamic scenarios, while outdoor tests focused on real-world UAV maneuverability and localization accuracy. Key findings from these evaluations are summarized below:

- Indoor static localization tests demonstrated an average RMSE of approximately 4.28 cm, confirming high measurement consistency.
- Indoor dynamic localization tests indicated accurate real-time tracking during controlled movements, providing confidence in the system's capability for motion tracking.
- Outdoor field tests conducted with a DJI Matrice 300 RTK drone further validated the system's performance, showing minimal localization errors and robust real-time tracking capability under realistic flight conditions.
- The machine learning model employing a MLP effectively reduced UWB measurement errors, significantly enhancing localization precision.
- Real-time data acquisition, processing, and communication were successfully integrated, underscoring the system's suitability for UAV applications requiring rapid and accurate positioning responses.

The comprehensive testing campaign at Boğaziçi University's Uçaksavar Campus provided additional validation of the system's operational effectiveness in real-world settings, complementing the earlier indoor test results. Nevertheless, some limitations

were identified, such as the lack of a ground truth system during indoor dynamic tests, restricting a quantitative assessment of performance metrics like RMSE and tracking errors.

To address identified limitations and further enhance system performance, several recommendations for future research are proposed:

- Implementation of a ground truth reference system in dynamic testing scenarios to quantify localization performance more precisely.
- Exploration of advanced signal processing or additional machine learning approaches to reduce multipath effects typical in UWB environments.
- Investigation into alternative ranging methodologies to increase update rates beyond the current TWR method.
- Adoption of high-quality IMU sensors to minimize sensor drift and improve orientation estimation accuracy.
- Transition to IEEE 802.15.4z-compliant UWB modules for enhanced performance and reduced multipath interference.
- Exploration of advanced state estimation techniques, including the Unscented Kalman Filter, Particle Filter, or novel Kalman filter variants, to effectively handle nonlinearities and uncertainties.

Implementing these recommendations will further enhance the system's accuracy, reliability, and operational robustness, contributing significantly to the advancement and practical adoption of autonomous UAV landing technologies.

REFERENCES

1. Jordan, S., J. Moore, S. Hovet, J. Box, J. Perry, K. Kirsche, D. Lewis and Z. T. H. Tse, “State-of-the-art Technologies for UAV Inspections”, *Radar, Sonar & Navigation*, Vol. 12, No. 2, pp. 151–164, 2018.
2. Tsouros, D. C., S. Bibi and P. G. Sarigiannidis, “A Review on UAV-Based Applications for Precision Agriculture”, *Information*, Vol. 10, No. 11, p. 349, 2019.
3. Li, Y., M. Liu and D. Jiang, “Application of Unmanned Aerial Vehicles in Logistics: A Literature Review”, *Sustainability*, Vol. 14, No. 21, p. 14473, 2022.
4. Erdelj, M., E. Natalizio, K. R. Chowdhury and I. F. Akyildiz, “Help from the Sky: Leveraging UAVs for Disaster Management”, *Pervasive Computing*, Vol. 16, No. 1, pp. 24–32, 2017.
5. Gyagenda, N., J. V. Hatilima, H. Roth and V. Zhmud, “A Review of GNSS-Independent UAV Navigation Techniques”, *Robotics and Autonomous Systems*, Vol. 152, p. 104069, 2022.
6. Lu, Y., Z. Xue, G.-S. Xia and L. Zhang, “A Survey on Vision-Based UAV Navigation”, *Geo-Spatial Information Science*, Vol. 21, No. 1, pp. 21–32, 2018.
7. Shastri, A., N. Valecha, E. Bashirov, H. Tataria, M. Lentmaier, F. Tufvesson, M. Rossi and P. Casari, “A Review of Millimeter Wave Device-Based Localization and Device-Free Sensing Technologies and Applications”, *Communications Surveys & Tutorials*, Vol. 24, No. 3, pp. 1708–1749, 2022.
8. Qiu, Z., Y. Lu and Z. Qiu, “Review of Ultrasonic Ranging Methods and Their Current Challenges”, *Micromachines*, Vol. 13, No. 4, p. 520, 2022.
9. Queralta, J. P., C. Martinez Almansa, F. Schiano, D. Floreano and T. Wester-

- lund, “UWB-Based System for UAV Localization in GNSS-Denied Environments: Characterization and Dataset”, *International Conference on Intelligent Robots and Systems (IROS)*, pp. 4521–4528, Las Vegas, NV, USA, 2020.
10. Shule, W., C. Martínez Almansa, J. Peña Queralta, Z. Zou and T. Westerlund, “UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems”, *Procedia Computer Science*, Vol. 175, pp. 357–364, 2020.
 11. Tiemann, J., F. Schweikowski and C. Wietfeld, “Design of an UWB Indoor-Positioning System for UAV Navigation in GNSS-Denied Environments”, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, Banff, Alberta, Canada, 2015.
 12. Tiemann, J. and C. Wietfeld, “Scalable and Precise Multi-UAV Indoor Navigation Using TDOA-Based UWB Localization”, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, Sapporo, Japan, 2017.
 13. You, W., F. Li, L. Liao and M. Huang, “Data Fusion of UWB and IMU Based on Unscented Kalman Filter for Indoor Localization of Quadrotor UAV”, *Access*, Vol. 8, pp. 64971–64981, 2020.
 14. Laidig, D. and T. Seel, “VQF: Highly Accurate IMU Orientation Estimation with Bias Estimation and Magnetic Disturbance Rejection”, *Information Fusion*, Vol. 91, pp. 187–204, 2023.
 15. Zeng, Q., Y. Jin, H. Yu and X. You, “A UAV Localization System Based on Double UWB Tags and IMU for Landing Platform”, *Sensors*, Vol. 23, No. 9, pp. 10100–10108, 2023.
 16. Li, J., Y. Bi, K. Li, K. Wang, F. Lin and B. M. Chen, “Accurate 3D Localization for MAV Swarms by UWB and IMU Fusion”, *14th International Conference on*

- Control and Automation (ICCA)*, pp. 100–105, Anchorage, AK, USA, 2018.
17. Strohmeier, M., T. Walter, J. Rothe and S. Montenegro, “Ultra-Wideband Based Pose Estimation for Small Unmanned Aerial Vehicles”, *Access*, Vol. 6, pp. 57526–57535, 2018.
 18. Ochoa-de Eribe-Landaberea, A., L. Zamora-Cadenas, O. Peñagaricano-Muñoa and I. Velez, “UWB and IMU-Based UAV’s Assistance System for Autonomous Landing on a Platform”, *Sensors*, Vol. 22, No. 6, p. 2347, 2022.
 19. Federal Communications Commission, *First Report and Order 02-48*, Tech. rep., Federal Communications Commission, 2002.
 20. Sahinoglu, Z., S. Gezici and I. Guvenc, *Ultra-Wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*, Cambridge University Press, Cambridge, UK, 2008.
 21. European Union, “Commission Implementing Decision (EU) 2024/1467 of 27 May 2024”, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024D1467>, 2024, accessed: December 19, 2024.
 22. Pozyx, “The State of UWB in 2024”, <https://www.pozyx.io/blog/the-state-of-uwband-in-2024>, 2024, accessed: December 19, 2024.
 23. Apple Support, “Ultra Wideband availability”, <https://support.apple.com/en-us/109512>, 2023, accessed: December 19, 2024.
 24. IEEE Standards Association, “IEEE Standard for Low-Rate Wireless Networks—Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques”, IEEE Std 802.15.4z-2020, 2020, accessed: December 19, 2024.
 25. Zion Market Research, “Ultra-Wideband Chipset Market by Application, End-

- User Industry, and Region: Global Industry Perspective, Comprehensive Analysis, and Forecast, 2023–2030”, <https://www.zionmarketresearch.com/report/ultra-wideband-chipset-market>, 2024, accessed: December 19, 2024.
26. FiRa Consortium, “FiRa Consortium: Making UWB Interoperable”, <https://www.firaconsortium.org/>, 2024, accessed: December 19, 2024.
 27. Omlox, “Omlox: The Open and Interoperable Standard for Real-Time Localization”, <https://omlox.com/>, 2024, accessed: December 19, 2024.
 28. Neiryneck, D., E. Luk and M. McLaughlin, “An Alternative Double-Sided Two-Way Ranging Method”, *13th Workshop on Positioning, Navigation and Communications (WPNC)*, pp. 1–6, Bremen, Germany, 2016.
 29. Inpixon, “UWB Localization: TDOA vs. TWR”, <https://www.inpixon.com/blog/uwb-localization-tdoa-vs-twr>, 2024, accessed: December 19, 2024.
 30. Oumar, O. A., M. F. Siyau and T. P. Sattar, “Comparison between MUSIC and ESPRIT Direction of Arrival Estimation Algorithms for Wireless Communication Systems”, *The First International Conference on Future Generation Communication Technologies*, pp. 99–103, London, UK, 2012.
 31. Yassin, A., Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs and E. Aboutanios, “Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications”, *Communications Surveys & Tutorials*, Vol. 19, No. 2, pp. 1327–1346, 2017.
 32. Krishnan, S., R. X. M. Santos, E. R. Yap and M. T. Zin, “Improving UWB Based Indoor Positioning in Industrial Environments Through Machine Learning”, *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1484–1488, Singapore, 2018.
 33. Nosrati, L., M. S. Fazel and M. Ghavami, “Improving Indoor Localization Using

- Mobile UWB Sensor and Deep Neural Networks”, *Access*, Vol. 10, pp. 20420–20431, 2022.
34. Si, M., Y. Wang, H. Siljak, C. Seow and H. Yang, “A Lightweight CIR-Based CNN With MLP for NLOS/LOS Identification in a UWB Positioning System”, *Communications Letters*, Vol. 27, No. 5, pp. 1332–1336, 2023.
 35. Radar Tutorial, “Multilateration (MLAT)”, <https://www.radartutorial.eu/02.basics/rp52.en.html>, 2024, accessed: December 19, 2024.
 36. Qorvo, “DWM1001-DEV Development Board”, <https://www.qorvo.com/products/p/DWM1001-DEV>, 2024, accessed: December 19, 2024.
 37. Nordic Semiconductor, “nRF52832 Bluetooth SoC”, <https://www.nordicsemi.com/Products/nRF52832>, 2024, accessed: December 19, 2024.
 38. Qorvo, “DWM1001C Module”, <https://www.qorvo.com/products/p/DWM1001C>, 2024, accessed: December 19, 2024.
 39. Qorvo, “DWM1000 Module”, <https://www.qorvo.com/products/p/DWM1000>, 2024, accessed: December 19, 2024.
 40. Decawave Ltd, *DW1000 Device Driver API Guide*, Dublin, Ireland, 2017, version 2.7.
 41. Decawave Ltd, *DW1000 User Manual*, Dublin, Ireland, 2017, version 2.18.
 42. Analog Devices, “ADXL345 Digital Accelerometer”, <https://www.analog.com/en/products/adxl345.html>, 2024, accessed: December 19, 2024.
 43. TDK InvenSense, “ITG-3205 Gyroscope Product Overview”, <https://invensense.tdk.com/products/itg-3205/>, 2024, accessed: December 19, 2024.

44. Honeywell, “HMC5883L 3-Axis Digital Compass IC”, <https://www.honeywell.com/en-us/products/technology/advanced-sensors/magnetic-field-sensors/hmc5883l>, 2024, accessed: December 19, 2024.
45. Benewake, “TFmini Plus LiDAR Sensor”, <https://en.benewake.com/TFminiPlus/index.html>, 2024, accessed: December 19, 2024.
46. Raspberry Pi Foundation, “Raspberry Pi 5”, <https://www.raspberrypi.com/products/raspberry-pi-5/>, 2024, accessed: December 19, 2024.
47. Sailboat Instruments, “Improved Magnetometer Calibration (Part 2)”, <https://sailboatinstruments.blogspot.com/2011/09/improved-magnetometer-calibration-part.html>, 2011, accessed: December 19, 2024.
48. Qorvo, “APS014 Antenna Delay Calibration in DW1000 Products and Systems”, <https://www.qorvo.com/products/d/aps014>, 2016, accessed: December 19, 2024.
49. Qorvo, “APS011 Sources of Error in TWR”, <https://www.qorvo.com/products/d/aps011>, 2016, accessed: December 19, 2024.
50. Branco, S., “CLARA – Embedded ML Tools”, <https://github.com/asergiobranco/clara>, 2020, accessed: December 19, 2024.
51. Despaux, F., A. Van Den Bossche, K. Jaffrès-Runser and T. Val, “N-TWR: An Accurate Time-of-Flight-Based N-Ary Ranging Protocol for Ultra-Wide Band”, *Ad Hoc Networks*, Vol. 79, pp. 1–19, 2018.
52. Hruška, O., “TinyFrame Library”, <https://github.com/MightyPork/TinyFrame>, 2024, accessed: December 19, 2024.
53. Levy, S. D., “TinyEKF: Lightweight C Implementation of the Extended Kalman

Filter”, <https://github.com/simondlevy/TinyEKF>, 2023, accessed: December 24, 2024.

