

UNIVERSITY OF CALIFORNIA  
SANTA CRUZ

**A STUDY OF ULTRA-WIDEBAND LOCALIZATION USING  
REVERSE TIME-DIFFERENCE OF ARRIVAL PRINCIPLES**

A dissertation submitted in partial satisfaction of the  
requirements for the degree of

BACHELOR OF ELECTRICAL ENGINEERING

in

ELECTRONICS AND OPTICS

by

**Nicholas S. Kuipers**

June 2024

The Dissertation of Nicholas S. Kuipers  
is approved:

---

Professor Steve McGuire, Chair

---

Professor Jason Eshraghian

---

Professor Dejan Milutinovic  
Undergraduate Director, Department of Electrical and Computer Engineering

Copyright © by

Nicholas S. Kuipers

2024

## **Abstract**

# A Study of Ultra-Wideband Localization using Reverse Time-Difference of Arrival Principles

by

Nicholas S. Kuipers

This thesis explores the characteristics, performance and accuracy of a Reverse Time-Difference of Arrival (RTDoA) localization model leveraging an Ultra-Wideband (UWB) radio system. By taking five Bitcraze Loco Positioning System (LPS) nodes and appending an RTDoA and hyperbolic localization scheme, this study empirically investigates time-difference of arrival (TDoA) calculation performance, localization efficiency, and measurement accuracy of a system built from commercially-available hardware, open-source software, and custom Python scripts under different scenarios including obstacles and multipath effects. This research contributes to further understanding of UWB and RTDoA performance, its strengths and shortcomings, and offers valuable insights for future studies aiming to optimize indoor positioning systems. Code is available at [https://github.com/Kuipman/uwb\\_tdoa](https://github.com/Kuipman/uwb_tdoa).

To my sisters

Morgain, Aeryn, Katelyn, and Abigail

for their endless love and support.

## Acknowledgments

First and foremost I extend my sincerest gratitude to my supervisor, Dr. Steve McGuire, for his continuous support and guidance throughout the course of this research. His words of encouragement and belief in me during times when I've doubted myself have truly been invaluable.

I am also deeply thankful to my reader, Professor Jason Eshraghian, for his critical feedback and constructive guidance on this project as well as others. My only regret is that I couldn't have taken up more of his valuable time.

My appreciation goes out to my colleagues and friends at the Human-Aware Robotics Exploration Lab for providing a strong, structured, and exciting work environment to grow and thrive in. Special thanks to Morgan Masters and Luca Altaffer for their constructive feedback and critical guidance during the course of writing this thesis.

I would also like to thank my family for all their love and encouragement. To my Dad and to Christa, to my sisters Morgain, Aeryn, Katelyn, and Abby, to my grandma Linda and to my aunt and uncle Di and Gar, thank you for your unending love and support and for always believing in me: I absolutely couldn't have done this without you.

Lastly, I can't leave out my friends outside the university who have stood by me for so many years and were essential in getting me to the finish line. To Jared and Kenny, thank you for always being a phone call away and for understanding when I

needed to skip out on our game nights to work on this. To Cecily, thank you for being my best friend and I genuinely wouldn't be here without you in my corner. To Eric, Mason, Jon, Jake, Big Chris, Little Chris and to everyone else: thank you for being genuine friends I've been able to count on for so many years and for always being there when I needed a laugh or a shoulder.

# Table of Contents

<b>Abstract</b>	<b>i</b>
<b>Dedication</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>3</b>
<b>3 Ultra-Wideband (UWB)</b>	<b>7</b>
3.1 Propagation Theory . . . . .	8
3.2 Ranging . . . . .	12
3.2.1 Two-Way Ranging . . . . .	15
3.2.2 Time-Difference of Arrival . . . . .	20
<b>4 TDoA Localization Experiment</b>	<b>27</b>
4.1 Hardware Setup . . . . .	28
4.2 Reverse TDoA Localization Algorithm . . . . .	31
4.2.1 Modified Two-Way Ranging . . . . .	31
4.2.2 Noise and Outlier Filtering . . . . .	38
4.2.3 Hyperbolic Localization: Nonlinear Least Squares . . . . .	39
<b>5 Results</b>	<b>44</b>
<b>6 Conclusion</b>	<b>52</b>
<b>References</b>	<b>54</b>

# List of Figures

1.1	Autonomous robotic systems are seeing wider use in precision agriculture and crop monitoring as valuable data collection tools . . . . .	2
3.1	Standard un-modulated radio wave at 5Hz versus a UWB pulse. Notice the UWB pulse is a few nanoseconds in length, typical for the communication protocol . . . . .	9
3.2	Spectra of Popular Communication Protocols. Here, UWB operates at very low power and exists close to the noise floor. Image courtesy of [1]	10
3.3	Depiction of the two-way ranging scheme. A minimum of three signals are sent between two UWB nodes to generate sufficient timestamps to calculate a Time of Flight measurement. . . . .	16
3.4	Example of Circular Localization. Here, a mobile tag's distances from three fixed anchors is determined and represented as a circle around each anchor. The area of intersection is the gap located just below coordinates (5,5), and the tag is estimated to be somewhere within this gap. . . . .	19
3.5	A single hyperbola generated from a TDoA measurement between a tag and two anchors. The tag is estimated to lie somewhere on one of the hyperbolic curves. . . . .	23
3.6	Multiple accurate TDoA measurements generate hyperbolas that intersect or nearly intersect at a point in space. The system estimates this point as the tag location (around coordinates (9.5,10) in this figure) . . .	23
4.1	TDoA experiment features four anchors of different heights at the four corners of a 4.5 meter by 4.5 meter square. The tag is placed within the convex hull of the square and connected to a laptop for localization calculations. To accurately model noise characteristics and localization performance the tag does not move for the duration of the experiment. . .	30
4.2	Sample packet received by the tag from anchor 3 during the experiment.	32

4.3	Diagram of the scheme for calculating the TDoA at the tag between two anchors. Here, a packet P1 is sent from A2 that is picked up by A1. A1 updates its recorded timestamps and ToF information before sending out a new packet P2. This is picked up by A2, that performs the same update and sends out a new packet P3. All three packets are picked up by the tag, and the respective timestamps are used to calculate the values necessary to determine the TDoA of this system. . . . .	33
4.4	2D Hyperbola vs 3D Hyperboloid . . . . .	40
5.1	Gaussians generated by calculated TDoA values for each pair . . . . .	46
5.2	Histograms of TDoA values for each pair, generated from a sample of approximately 70,000 packets . . . . .	47
5.3	Top-view visualization of the nonlinear least squares position estimation to best show the curves of the hyperboloids and their intersect point . .	49
5.4	Side-view visualization of the nonlinear least squares position estimation. Note the X and Y coordinates of the estimation (yellow dot) are extremely close to the true position (purple dot) while the Z coordinate has a considerable offset . . . . .	50

# Chapter 1

## Introduction

In recent years the field of autonomous robotic systems has experienced remarkable advancements, transforming what was once considered science fiction into real, practical tools. These systems, capable of performing complex tasks with little-to-no human intervention, have made significant strides in various domains including manufacturing, healthcare, environmental monitoring, and precision agriculture (see Figure 1.1). The broader industry's latest emphasis on artificial intelligence, machine learning, and real-time data processing have helped fuel this rapid progress by enabling autonomous robots to operate with increasing precision, efficiency, and adaptability while alleviating or expediting tasks and labor previously reserved for human specialists.

Critical to autonomous operations is the field of navigation. There are different phases associated with navigation including perception, cognition, motion control, and localization [2], all of which contribute to a robot's ability to understand their spatial context and accurately determine their paths of travel and operation without



(a) Aerial Agricultural Survey at UCSC



(b) Robot Dog Picking Some Fruit

Figure 1.1: Autonomous robotic systems are seeing wider use in precision agriculture and crop monitoring as valuable data collection tools

human intervention. The latter phase, *localization*, enables a robot to determine their current position and orientation with precision, and is a crucial facet of any autonomous algorithm planning its next steps.

To accomplish localization autonomous technologies integrate various sensors, algorithms, and data processing techniques to provide real-time position and orientation information. There is no "one-size fits all" to the localization problem and many solutions exist for many types of robots and environments. This is a complex but critical problem to solve, as accurate localization allows a robotic system to effectively maneuver through complex and dynamic environments, interact reliably with objects and humans, plan routes, and perform tasks without human intervention.

# Chapter 2

## Background

Localization alone cannot be performed without appropriate environmental data to infer position and orientation from. This data is usually generated by sensors perceiving the robot and its environment in a myriad of ways, and many popular sensor technologies exist today to enable autonomous systems to reliably perceive and navigate their environments. *Inertial Measurement Units* (IMUs) are devices that use accelerometers and gyroscopes to provide real-time data on changes in velocity and orientation, and see widespread use. Off-the-shelf IMUs are generally inexpensive, small, low-power, and do not add significant complexity to a system. However, cheaper IMUs may generate less-reliable, less-precise measurements. Further, these suffer from accumulating measurement drift, which can drastically affect or outright break localization calculations over time. Through IMU data alone there are methods to estimate a robot's current position based on the data and the bot's previously-known position through a process called *dead reckoning*[3, 4]. However, to account for accumulating drift it is

more popular to use IMUs as part of a broader sensor fusion [5, 6].

*Light Detection and Ranging* (LiDAR) sensors are active sensors that use laser pulses to measure distances to objects, and have been used to create high-resolution 3D maps of an environment. LiDAR offers significant benefits in accurate localization and real-time mapping and are heavily explored in the autonomous vehicles space, becoming a *de facto* standard for the industry [7]. However, these sensors are usually very expensive, suffer significantly from environmental conditions such as fog and reflective surfaces, and come with substantial power and data processing requirements that restrict their use in smaller robotic systems [8]. *Visual odometry* (VO), comparatively, is able to leverage inexpensive camera systems (some costing as little as 25 dollars [9]) to offer robust localization solutions using only captured images. These also carry an added benefit as less power-hungry *passive* sensors, since unlike LiDAR these visual sensors do not project any kind of signals out into the operational space. However, the performance of these sensors is still vulnerable to environmental factors such as shadows, glare, changing features and/or featureless areas, which are all especially present outdoors [10]. Further, pushing camera images through a localization algorithm such as feature detection usually requires the additional use of dynamically building a map or using a pre-built map [11], requiring significant processing power and limiting real-time localization for smaller systems.

The *Global Positioning System* (GPS) has been a revolutionary tool for navigational applications by enabling real-time location tracking for personal, commercial, and military purposes. The GPS operates via a series of signals sent from a constella-

tion of satellites to the Earth’s surface that are used by a receiving device to calculate its position on Earth by trilateration, which in theory grants navigational capability to a receiver in any area visible to the open sky. However, the accuracy of GPS in ideal conditions generally approaches a minimum of five meters, and can be further adversely affected by signal blockages from trees, mountains, atmospheric conditions, and satellite geometry [12, 13]. For this reason, precision operations such as environmental and agricultural surveys often augment GPS with additional systems such as *Rapid static* or *Real-Time Kinematic* systems [14] to enable centimeter-accurate measurements or better, at the cost of significant system complexity. Further, GPS signals encounter high levels of attenuation inside buildings due to a large number of obstructions and sources of interference in close proximity to a receiver, making indoor GPS navigation unreliable [15]. Despite these drawbacks, however, GPS systems are now widely used in the navigation and localization space as a standalone system or as part of a broader sensor suite primarily due to their ease-of-access. This is less applicable, however, in environments such as indoor spaces, mountain ranges, dense forests, and other areas where GPS is infeasible (collectively referred to hereafter as ”GPS-denied Environments”) hence accessible alternative localization methods are crucial to explore.

This thesis is an expansion on the idea of GPS-denied navigation through an exploration in a Time-Difference of Arrival (TDoA) technique using Ultra-Wideband (UWB) radio communications. UWB is a wireless technology whose pulse-based propagation method can be leveraged to calculate precise Time-of-Flight (ToF) calculations of a signal, and by extension a mobile receiver’s position with centimeter-level accuracy.

The intention is to produce a proof-of-concept for a low-cost, low-power localization system that can reliably operate in GPS-denied environments with precision suitable for autonomous robotic operations. This thesis first explores the propagation characteristics behind UWB and the primary ranging and localization methods used with this technology, with an enhanced focus on the Time-Difference of Arrival scheme. Then, a proof-of-concept for a Reverse Time-Difference of Arrival (RTDoA) method is demonstrated using a physical system, and its noise profile is analyzed. Finally, potential future iterations to the system and broader implementations of RTDoA localization are discussed.

# Chapter 3

## Ultra-Wideband (UWB)

Ultra-Wideband (UWB) communication is a wireless technology characterized by its ability to transmit data over a wide frequency spectrum, typically within the 3.1 GHz to 10.6 GHz range. The name *ultra-wideband* is derived from this spectrum, operating with a bandwidth several times larger than traditional narrowband and wideband communication systems that generally operate within a more confined range rarely exceeding 500 MHz. This broad spectrum at the high end of the communication spectrum leads to multiple benefits including the delivery of high data rates exceeding 1 Gbps while maintaining low power consumption, making UWB an appealing choice for various applications including short-range wireless communication, radar, and imaging. Utilizing a low power footprint on a wide spectrum also allows the protocol to operate close to the noise floor (see Fig 3.2 for more details), projecting minimum interference with other conventional communications protocols.

UWB transmits pulses that, due to their sufficiently-high frequency spectrum,

are generated with a high temporal resolution. This paves the way for advanced wireless applications including precise positioning and tracking a transceiver with centimeter-level accuracy. This section describes this pulse-based communication at a high level, discusses its applications for precise positioning, and details two popular ranging techniques UWB is leveraged to employ.

### 3.1 Propagation Theory

Typical UWB communication is pulse-based, using a combination of the Pulse Position Modulation (PPM) and Binary Phase Shift Keying (BPSK) techniques to encode binary information (0 or 1 digits) in the position and orientation of a pulse or pulses within a predetermined time period. PPM encodes data by varying the position of the pulse along a given time frame, and benefits from UWB's short-duration, precise pulses by allowing the receiver to distinguish between different pulse positions accurately. This is especially beneficial in environments with excessive noise and in cases of multipath interference, a common issue in wireless communication where signals reflect from surfaces in the environment and create delayed copies at the receiver that interfere with the primary signal. Comparatively, BPSK encodes data by shifting the phase of the pulse by either 0 or 180 degrees for a digit 0 or 1 respectively, and is similarly resistant to amplitude variations and noise present in a real-world system. The result is a rapid pulse oscillating between 0 (a negative value) and 1 (a positive value) around a neutral threshold and resistant to noise and sources of interference. This is illustrated in Fig-

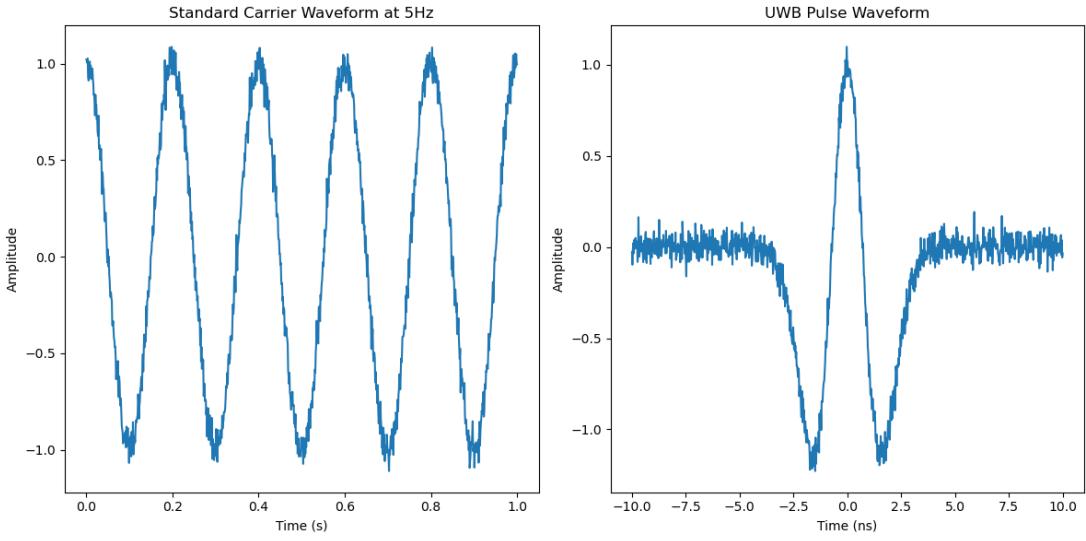


Figure 3.1: Standard un-modulated radio wave at 5Hz versus a UWB pulse. Notice the UWB pulse is a few nanoseconds in length, typical for the communication protocol

ure 3.1 alongside a sinusoidal, un-modulated carrier wave typical for many narrowband systems.

UWB operates in the gigahertz ranges of the frequency spectrum meaning generated pulses are optimally nanoseconds in length, allowing for many pulses to be sent over a short time frame generating extremely fast data transfer speeds. These pulses are spread over a very wide (perhaps an ultra-wide) frequency spectrum, meaning the energy of each pulse is also distributed over this wide frequency band. This has the positive consequence of UWB being able to operate with a very low *power spectral density* (PSD), producing signals that operate just above the communications noise floor and drastically reducing the power requirements for UWB devices. The result is a protocol that generates minimal interference (see Figure 3.2) to other narrowband and

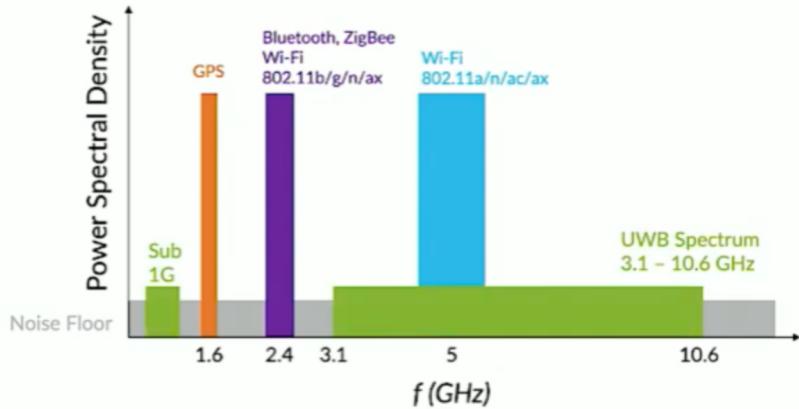


Figure 3.2: Spectra of Popular Communication Protocols. Here, UWB operates at very low power and exists close to the noise floor. Image courtesy of [1]

wideband communications protocols [16].

A notable application of UWB is in precise position estimation: its high time resolution allows UWB to provide distance measurements with centimeter-level accuracy by calculating the time of flight of the pulses. This performs significantly better for localization than other wireless localization technologies such as Wi-Fi or Bluetooth, and is leveraged in various applications including indoor positioning systems [6, 17, 18, 19], asset tracking [20], and in extracurricular research applications such as tracking the movement of soccer players over a season [21].

Thanks to various industry and governmental bodies UWB technology enjoys a fairly robust standardization: for instance the IEEE 802.15.4a standard defines a UWB physical layer for wireless personal area networks [22], which supports applications requiring high precision ranging and low power consumption. This has paved the way for

many interoperable UWB solutions from different manufacturers to be adopted across different industries and applications, and has contributed to the growing research on UWB applications.

The regulations on UWB do not end with IEEE standards: while UWB's low-power operations generate minimal interference it is important to note that interference with narrowband services is still possible. Critical services including WiFi, wireless channels for air traffic control, and GPS operate on bandwidths contained within the UWB spectrum, risking spectral overlapping between UWB and any of these services. As a consequence the Federal Communications Commission (FCC) has established strict regulations on UWB, limiting its applications to ensure the protocol can coexist with existing wireless systems [23]. It turns out UWB's low PSD is by design, thanks to a key FCC regulation limiting the protocol's total PSD to less than -41.3 dBm/MHz (a very small value) minimizing the risk of interference with other radio systems. In [23] the FCC categorizes UWB devices into different classes with different frequency bands and power limits based on their intended applications and operating environments. These classes range from imaging systems with more generous frequency spectra such as ground-penetrating radar [24], and indoor/outdoor communication systems and surveillance systems that come with more stringent power limits. The FCC also instituted regulations on aircraft employing Ultra-Wideband communications, an important consideration that will be revisited in the experiment section of this thesis.

## 3.2 Ranging

*Ranging* is a process in which a signal is sent from one point in space to another to determine the distance between them; usually this is calculated through measuring the time it takes for the signal to travel (called the *Time of Flight* or ToF) multiplied by the speed of the signal. Ranging can actually be quantified by a somewhat-common practice in the United States: estimating one's distance (roughly) from a thunderstorm by counting the seconds between seeing a bolt of lightning and hearing the thunder. While the light will reach an observer almost instantly a sound wave takes around five seconds to travel one mile, so in practice an individual can count the seconds between the flash of lightning and roll of thunder, then divide by five to determine their distance in miles from the storm.

In wireless communication systems this calculation typically involves sending one signal from a transmitter to a receiver; the time of flight is the duration the electromagnetic wave takes to travel from each point, and this wave travels at the speed of light. The distance between these points, then, can be calculated using equation 3.1.

$$d = c * \text{ToF} \quad (3.1)$$

where  $c$  is the speed of light, or  $3 * 10^8$  m/s.

As a brief example of a generic distance calculation say a wireless signal has a ToF of one microsecond to travel from a transmitter to receiver. By equation 3.2:

$$d = (3 * 10^8) * (1 * 10^{-6}) \quad (3.2)$$

resulting in a distance  $d = 300$  meters between the transmitter and receiver. Going further a ToF of a nanosecond corresponds to a 0.3 meter distance, or 30 centimeters. While 30 centimeters to a nanosecond certainly demonstrates the sheer speed of light, this also highlights an essential component of wireless ranging systems: clock synchronization.

For calculating the distance between a lightning strike and an observer considerations including an extremely accurate stopwatch and micro-estimations of the length of time for the light to reach the eye, for the brain to process the light and hit the stopwatch, for the stopwatch to register the button press and begin and stop recording, are not necessary. It is usually enough in this circumstance for an observer to hit their stopwatch at around six seconds and deduce "that strike was a bit over a mile away from me" regardless of whether the true distance is 1.2 miles or 1.24335543 miles away. In ranging applications with electromagnetic waves traveling at the speed of light, however, calculating a precise ToF becomes far more important as a time offset of mere nanoseconds can generate a distance several meters or more off from the true distance, making many robotic localization applications infeasible.

In the radio localization space this problem is the central focus of *precision time synchronization*, a rich field of research with ample forays made in Ultra-Wideband localization specifically [25]. UWB's unique propagation method (nanosecond-length pulses spread over a wide frequency band) generates an extremely-fine time resolution

between pulses that can be leveraged at both a transmitter and receiver of a signal to calculate ToF within the range of nanoseconds to sub-nanoseconds. To accomplish this each UWB node is equipped with a high frequency clock (usually with speeds of tens of megahertz) that undergo multiplications in software to generate nanosecond-scale clock speeds. In addition to generating cycle speeds necessary for rapid UWB pulses, the UWB clock's value is used to track and record a *timestamp*, or precise point in time at which a signal is either transmitted from a source or arrives at a receiver. It is with timestamp calculations that UWB is able to calculate ranging measurements: by calculating the difference in the timestamps recorded when a signal was transmitted from one point and received at another point an extremely-accurate time of flight can be deduced.

Naturally, ranging still isn't quite this easy as 100% accurate clock synchronization isn't yet possible: typical wireless communications feature two devices with two separate clocks operating asynchronously. These clocks may even be manufactured to tick at the exact same frequency but will experience different levels of frequency drift due to imperfections in the clock crystal, which can cause major accumulating offsets for nanosecond-scale operations. Even with wireless synchronization methods such as in [25] a given node can only ultimately guess the true clock value of a distant node. To that effect, a timestamp can be recorded at the transmitter and receiver, but there is no assurance a calculated time of flight value will generate a coherent value, let alone a value near the true ToF.

Several methods have been developed to resolve this problem. By far, the two

most popular ranging methods for UWB are: Two-Way Ranging (TWR) and Time-Difference of Arrival (TDoA). These methods are popular due to their straightforward design and implementation, and have seen many iterations in the UWB localization space [26, 17, 19, 18, 21, 20, 27, 28, 29, 30].

### 3.2.1 Two-Way Ranging

Two-Way Ranging (TWR) is a direct method to accurately calculate the time of flight of a signal without the need for clock synchronization between two UWB devices. For this method (see Figure 3.3) both devices act as a transmitter and as a receiver sending multiple signals between each other to calculate the *round-trip* time of flight. The resulting time of flight is calculated at the receiving node of the initial signal, which optionally can send that calculation back to the initial transmitter.

For Figure 3.3 assume a mobile UWB node (hereafter referred to as a "tag") wants to determine its distance to a nearby stationary UWB node (hereafter referred to as an "anchor"). The tag sends out an initial *poll* message containing its identifying information and records the precise timestamp  $T_{SP}$  at which the poll message was sent according to the tag's local clock. The anchor picks up the poll message and records the timestamp  $T_{RP}$  at the precise moment of receipt. The signal is processed at the anchor and a new *response* message containing the anchor's identifying information is sent, with the anchor recording the sending timestamp  $T_{SR}$  with reference to its own local clock. The tag receives this response message and records the appropriate timestamp  $T_{RR}$ . After another processing period the tag prepares a *final* message containing its

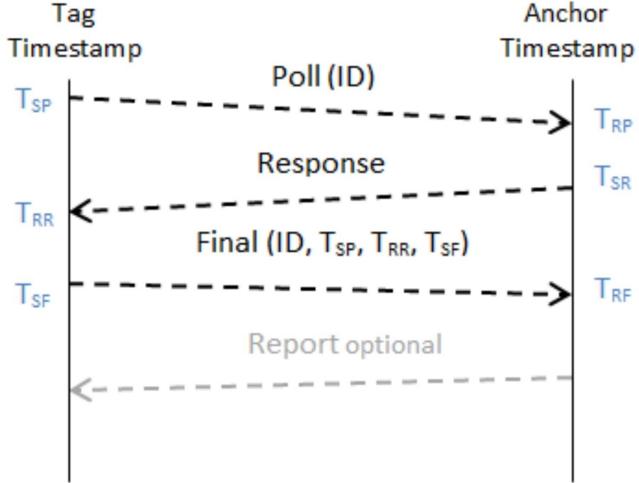


Figure 3.3: Depiction of the two-way ranging scheme. A minimum of three signals are sent between two UWB nodes to generate sufficient timestamps to calculate a Time of Flight measurement.

identifying information, the sending timestamp of the final message  $T_{SF}$ , and the two timestamps previously recorded by the tag ( $T_{SP}, T_{RR}$ ).

At the anchor, once the final message is received the appropriate timestamp  $T_{RF}$  is recorded. From here, the round-trip time of flight can be calculated by considering the time for one node to complete a message exchange (transmit one message and receive a response) with the other node. Then, since the result is effectively the elapsed time for the signal to travel the distance twice, this is divided by 2.

For this system of three messages there are two complete exchanges that can be considered: Poll to Response and Response to Final. Consider the Poll to Response exchange: there are two timestamps  $T_{SP}, T_{RR}$  recorded at the tag and two timestamps  $T_{RP}, T_{SR}$  recorded at the anchor. If the processing time at the anchor were instantaneous

the round-trip time of flight would simply be the difference between when the first signal is sent and when the second signal is received at the tag.

$$RToF_{PR}(\text{ideal}) = T_{RR} - T_{SP} \quad (3.3)$$

Since both of these timestamps are recorded at the tag no clock synchronization with the anchor is necessary. However, in the real-world a receiver takes time to pick up and process the poll message, then construct and transmit the response message. This processing time must be considered. The timestamps  $T_{RP}, T_{SR}$  are thus recorded to approximate this processing time, which must be subtracted from equation 3.3. The actual round-trip time of flight, then, is:

$$RToF_{PR}(\text{actual}) = (T_{RR} - T_{SP}) - (T_{SR} - T_{RP}) \quad (3.4)$$

The round-trip time of flight is simply the time it takes for the signal to travel between two points twice. So, the time of flight:

$$ToF_{PR} = (RToF_{PR})/2 = ((T_{RR} - T_{SP}) - (T_{SR} - T_{RP}))/2 \quad (3.5)$$

For greater accuracy the Response to Final exchange is also considered and added to 3.5. The round-trip time of flight for this exchange follows the same methods used to find equation 3.5, leading to:

$$ToF_{RF} = (RToF_{RF})/2 = ((T_{RF} - T_{SR}) - (T_{SF} - T_{RR}))/2 \quad (3.6)$$

Therefore, the final time of flight calculation for a TWR system is:

$$\begin{aligned} ToF &= (ToF_{PR} + ToF_{RF})/2 \\ &= ((T_{RR} - T_{SP}) - (T_{SR} - T_{RP}) + (T_{RF} - T_{SR}) - (T_{SF} - T_{RR}))/4 \end{aligned} \quad (3.7)$$

Once the time of flight is found the anchor can calculate its distance from the mobile tag by multiplying ToF by the speed of light (see equation 3.1). This alone is not sufficient to determine an accurate position, however if the tag's distance from other fixed anchors can also be determined there are several localization methods that can be employed including *triangulation*, *trilateration*, *circular* or *spherical localization* (see Figure 3.4), and *elliptical localization*. These localization algorithms are outside the scope of this thesis, however curious readers are encouraged to review [29, 27] which report successful applications of UWB TWR trilateration and elliptical localization.

The largest advantage to two-way ranging is its ease of implementation. There is no direct arithmetic between two timestamps that are from different devices with different clocks, meaning precise clock synchronization is unnecessary (it's worth mentioning extra arithmetic is needed if the two clocks do not operate on the same frequency). The message packets are small and easy to design in software. Most popular UWB devices on the market such as the DW1000 [31] are manufactured with the hardware capability to record precise timestamps of message receptions and transmissions for ready transfer via SPI to a microcontroller or computer. Asset tracking systems and indoor positioning of a single or few items is easy to implement and put out quickly

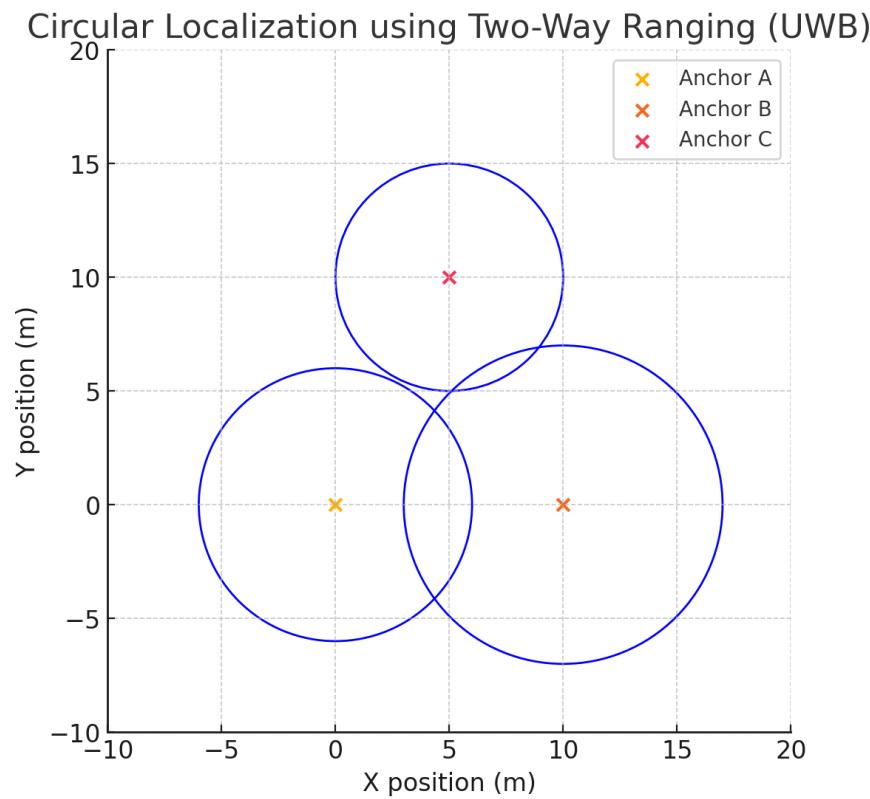


Figure 3.4: Example of Circular Localization. Here, a mobile tag's distances from three fixed anchors is determined and represented as a circle around each anchor. The area of intersection is the gap located just below coordinates (5,5), and the tag is estimated to be somewhere within this gap.

with a TWR system.

An immediate drawback to two-way ranging is that it is very RF-heavy, requiring lots of message transmissions to generate up-to-date ToF calculation. This is a concern for two main reasons: first this system is challenging to scale as the number of signals in the air and processing power needed at the base stations increase exponentially as more tags are added into the system. The second reason draws back to the previous discussion on FCC regulations: the agency states in [23] that any type of UWB transmission from an aerial device is prohibited outdoors to prevent interference with other aircraft channels. This completely negates the capability for two-way ranging to be employed on a mobile aerial device – for these types of applications other ranging methods must be considered.

### 3.2.2 Time-Difference of Arrival

Time-Difference of Arrival (TDoA) is a popular technique for UWB ranging, and is the focus of this thesis. Unlike TWR, which requires both the anchor and tag to exchange message packets, TDoA designates the tag as solely a transmitter or receiver: anchors can still readily exchange information among themselves, but typically communication with the tag is unidirectional. Consequentially, the term *one-way ranging* is often used interchangeably with TDoA [18, 19]. This approach has reduced system complexity versus TWR at the tag and carries greater potential for scalability with additional tags, making TDoA a valuable option for precision localization [17].

In traditional TDoA setups, a single tag with an unknown position periodically

transmits a packet that includes the tag’s identifying information along with the packet’s transmission timestamp. This packet is received by multiple anchors positioned at known, fixed locations within the tag’s range. Each anchor records the timestamp of the precise moment it receives the packet – through intercommunication between the anchors (usually through a central hub) these timestamps are compared to calculate the differences in the times of arrival, or the *time-difference of arrival* of the signal at various pairs of anchors. Importantly TDoA is **not** equivalent to the time of flight measurements in TWR, but can better be described as the *difference* between two ToF values. This corresponds to different methods of localization needing to be employed than those of TWR.

Consider a simple two-dimensional (2D) system with a single transmitting tag and two anchors. Let the tag  $T$  at unknown position  $(x_t, y_t)$  transmit a packet at time  $t_o$  to anchors  $A_1$  and  $A_2$  at known coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$ . The true distances from the tag to the anchors are  $d_1$  and  $d_2$  respectively, and the speed of light is  $c = 3 \times 10^8 \text{ m/s}$ . The time it takes for the signal to reach the two anchors is

$$t_1 = t_o + \frac{d_1}{c} \text{ and } t_2 = t_o + \frac{d_2}{c} \quad (3.8)$$

then, the difference in arrival times at the two anchors is

$$\Delta t = t_2 - t_1 = \left(t_o + \frac{d_2}{c}\right) - \left(t_o + \frac{d_1}{c}\right) = \frac{d_2 - d_1}{c} \quad (3.9)$$

This can be simplified to the shorthand TDoA calculation:

$$c\Delta t = d_2 - d_1 \quad (3.10)$$

The distance values  $d_1$  and  $d_2$  can be expressed using the Euclidean Norm

$$||d_1|| = \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2} \text{ and } ||d_2|| = \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2} \quad (3.11)$$

Substituting these into 3.10 gives

$$c\Delta t = \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2} - \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2} \quad (3.12)$$

Equation 3.12 describes a hyperbola with foci at the anchor positions  $x_1, y_1$  and  $x_2, y_2$ . The curve of the hyperbola represents all possible locations that can generate the calculated TDoA measurement: the tag's position  $x_t, y_t$  will lie somewhere on this hyperbola. The sign of  $c\Delta t$  helpfully indicates which curve the tag sits upon: a positive value implies that  $d_1 < d_2$  and the tag is closer to  $A_1$  than  $A_2$ . The opposite is true when  $c\Delta t$  is negative.

This hyperbola can be leveraged, appropriately, with *hyperbolic localization*, a method that is very popular in TDoA applications [17, 18, 19, 30]. Of course, as with TWR, a single measurement between two anchors and a tag is not enough to generate a unique localization solution: the tag can lie anywhere on the generated hyperbolic curve. A second hyperbola generated by a second anchor pair (usually comprised of one of the anchors from the first pair and a new anchor) will generally intersect the first hyperbola,

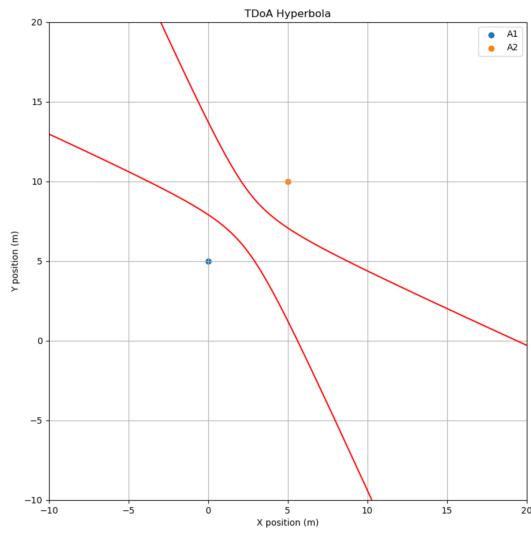


Figure 3.5: A single hyperbola generated from a TDoA measurement between a tag and two anchors. The tag is estimated to lie somewhere on one of the hyperbolic curves.

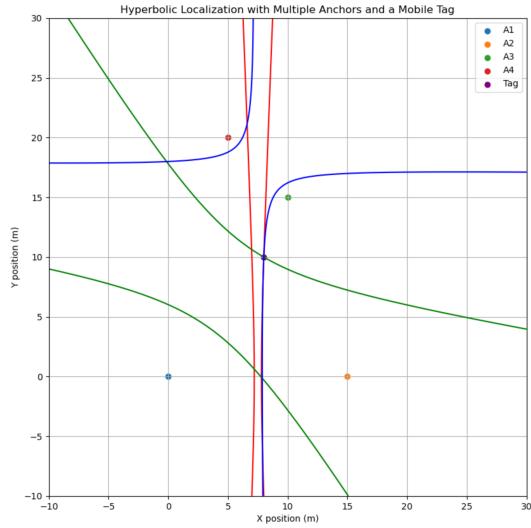


Figure 3.6: Multiple accurate TDoA measurements generate hyperbolas that intersect or nearly intersect at a point in space. The system estimates this point as the tag location (around coordinates (9.5,10) in this figure)

providing a unique point in 2D space where the tag is likely to be. Similarly, for 3D space a three-anchor system will not be sufficient since it generates a flat plane where the tag can lie at any point – a fourth anchor is required for accurate 3D positioning. To find this intersect point many estimation methods including *nonlinear least squares regression* [32] and the *Multivariate Newton's Method* [33] can be employed.

TDoA carries many advantages over TWR, namely scalability: TDoA schemes are significantly-less RF-heavy, as only a single transmission from the mobile tag is sufficient for the anchors to calculate a position estimate (optionally a return message containing this position estimate can be sent back to the tag). More tags, then, can be added to the system without much additional processing overhead. Conversely, the TDoA scheme requires additional system complexity, namely a method for the anchors to communicate (either wired or wirelessly) between each other to calculate a position estimate. Traditionally, this is done with a primary anchor receiving measurements from all other anchors and calculating the tag's position.

Unlike TWR, clock synchronization must be addressed in any implementation of TDoA. For this scheme it is critical the base stations operate on the same clock value to accurately calculate the TDoA of received signals from the tag. Since the base stations are almost always static and unmoving this is usually accomplished by hard-wiring the base stations to a single reference clock source, whether that be a standalone high-precision clock or one of the base stations' onboard clocks. The tag, helpfully, is *not* required to maintain synchronization with the base stations since the reception times and not the transmission times of its signals are used for TDoA calculations.

TDoA as described thus far is a system in which the mobile tag sends repeated transmissions that are picked up by stationary anchors to calculate its position. A major benefit to standard TDoA is that clock synchronization at the tag can be avoided, however a transmit-only tag system retains the issue of FCC regulations negating its use as an onboard system for an unmanned aerial vehicle (UAV). To safely utilize a UWB localization system aboard a UAV, the tag cannot actively transmit *any* UWB signals.

Reverse TDoA (RTDoA) is, as the name implies, the reverse of traditional TDoA: the tag of unknown position acts a *receive-only node* picking periodic transmissions from a series of anchors of known, unchanging positions. Computational complexity at the base stations is significantly reduced in this scheme as calculations are now almost entirely ran at the tag: the tag receives and compares packets from the base stations to calculate the TDoA of these signals, and updates its position through an onboard localization algorithm. The tag can then optionally communicate its new calculated position (through a non-UWB method) back to the base stations.

RTDoA effectively solves the FCC regulation issue by negating the requirement for a UAV to transmit UWB messages. Further, the mathematical concepts employed in RTDoA are identical to traditional TDoA; their differences lie purely in their implementation. That said, the largest challenge with RTDoA is clock synchronization at the base stations *and* at the tag. This is a hard requirement: timestamps and times of flight are calculated directly at the tag and rely on the tag's local clock. Efforts have been made to solve this problem by continuously synchronizing the tag's clock to the

base stations through wireless methods [25, 34], however these are difficult to implement and rely on additional, costly equipment. Alternatively, the method conducted in [17] uses a much simpler method of conducting a TWR scheme among the base stations, which exchange their timestamping information, and having the mobile tag pick up these signals and parse timestamp values in accordance with its local clock through some straightforward arithmetic. This method does not require clock synchronization and is simple to implement and understand, therefore this is the RTDoA method from which the experiment in this thesis is inspired.

# Chapter 4

## TDoA Localization Experiment

The objective of this experiment is to construct a Reverse Time-Difference of Arrival (RTDoA) system in which four anchors (the minimum required for 3D localization) broadcast packets that are intercepted by a mobile tag, and establish a proof-of-concept for localization within the convex hull established by the layout of the anchors in the system. The RTDoA system is constructed using off-the-shelf UWB radios operating in a large indoor environment with a combination of open-source firmware and custom Python scripts. The tag is *stationary* and does not move through the course of the experiment: the desired result is a stable and accurate location estimate of the tag, as well as generating an approximate noise profile in the tag's TDoA measurements as a consequence of wireless communications.

## 4.1 Hardware Setup

The experiment makes use of five Bitcraze Loco Positioning System (LPS) UWB nodes, depicted in Figure 4.1a. Each of these nodes feature an STM32 microprocessor and a DW1000 UWB integrated chip (IC) designed to run the UWB protocol in accordance with IEEE standard 802.15.4-2011 and features a transceiver and omnidirectional antenna. Four of these nodes act as stationary anchors arranged at the corners of a 4.5 meter by 4.5 meter square and raised above the ground by a tripod with a height of 1 to 3 meters (see Figure 4.1b). The fifth node acts as the mobile tag and lies at some unknown location within the drawn square, or *convex hull* of the anchors, positioned on a tripod at some height above the ground. The tag intercepts transmitted packets from the surrounding anchors and outputs this information to a laptop computer that performs the TDoA measurements and hyperbolic localization calculations. Figures 4.1c and 4.1d depicts this hardware setup.

The anchors are programmed with a *modified two-way ranging* scheme provided in the LPS firmware [35] in which their transmitted packets are not targeted to a specific receiver but are broadcast to all other anchors. These packets (a sample is shown in Figure 4.2) contain much more data than standard TWR packets and include recent timestamp and time of flight information from all other anchors the transmitting station has recently exchanged packets with. This modified scheme is for the benefit of the mobile tag, which is able to intercept these packets and parse the timestamp data critical to its TDoA calculations. To reiterate, in this scheme there is *no* clock synchronization

between any of the nodes – instead a straightforward algorithm considering each unsynchronized clock is used at the tag to estimate remote timestamp values with the tag’s clock reference through standard arithmetic.



(a) Bitcraze Loco Positioning System (LPS) Node



(b) LPS Node attached to tripod with custom mount



(c) Experiment Setup, Far View



(d) Tag Connection to Laptop

Figure 4.1: TDoA experiment features four anchors of different heights at the four corners of a 4.5 meter by 4.5 meter square. The tag is placed within the convex hull of the square and connected to a laptop for localization calculations. To accurately model noise characteristics and localization performance the tag does not move for the duration of the experiment.

## 4.2 Reverse TDoA Localization Algorithm

To reiterate: Reverse TDoA is the difference in time of flight between signals from two anchors received at the tag. For this method the tag operates as a receive-only device "sniffing" for message packets exchanged by the stationary anchors in a modified TWR scheme. For a system with four anchors six unique TDoA values can be calculated and used as part of a hyperbolic localization algorithm to estimate the tag's current position in 3D space.

### 4.2.1 Modified Two-Way Ranging

Consider a sample system where anchor 1 (A1) and anchor 2 (A2) are communicating in the modified TWR scheme and observed by the tag. This system is detailed in Figure 4.3 and this figure will be referenced often for this section. Anchor 2 (A2) begins by broadcasting a packet  $P_1$  that is picked up by anchor 1 (A1). A1 parses this bit stream and updates its timestamp and ToF information between itself and A2 before broadcasting its own packet  $P_2$ . This new packet is received by A2, which performs the same steps before broadcasting a third packet  $P_3$ . Note this TWR protocol is *continuous*:  $P_2$  and  $P_3$  can be used as part of a separate ToF calculation with a fourth packet  $P_4$ ,  $P_3$  and  $P_4$  can be used with  $P_5$ , and so on. Since these packets are all *broadcast* packets and not targeted towards a particular destination the central idea behind this scheme is that every packet transmission allows all receiving anchors to update their time of flight data with the transmitting anchor. By extension, with every

```

data: !!binary |
  MCoAJrWgAwTweDN1dVuEAe9lug5nYoQCi5MST1nhhQ==
from: 3
remoteAnchorData:
- distance: 33883
  id: 4
  rxTimeStamp: 1970615160
  seq: 112
- distance: 33890
  id: 1
  rxTimeStamp: 1729018466
  seq: 111
- distance: 34273
  id: 2
  rxTimeStamp: 1498354323
  seq: 11
remoteCount: 3
rxSys: 1713225045.9036129
seq: 42
to: 255
tof:
  1: 4.403881235288964
  2: 6.2008268390988235
  4: 4.371038887438658
ts: 311278726269
txTimeStamp: 2696226304
type: 48

```

Figure 4.2: Sample packet received by the tag from anchor 3 during the experiment.

received packet the mobile tag is also able to update its own TDoA calculation for A1 and A2. By intercepting these packet exchanges between all anchors in the system and calculating multiple TDoA values the tag is able to reliably and consistently update its position estimation.

In this modified TWR scheme the message packets contain significantly more information for the benefit of the tag. Figure 4.2 shows a packet received by the tag from anchor 3 (A3) during the experiment. Here, the *remoteAnchorData* section contains the time of flight (in clock ticks) between A3 and each of the other *remote* anchors in the system. These ToF values are generated from packets previously-received from these remote anchors at A3. Also included is the sequence number and receive timestamp for the last packet A3 received from each remote anchor. The latter half of this packet

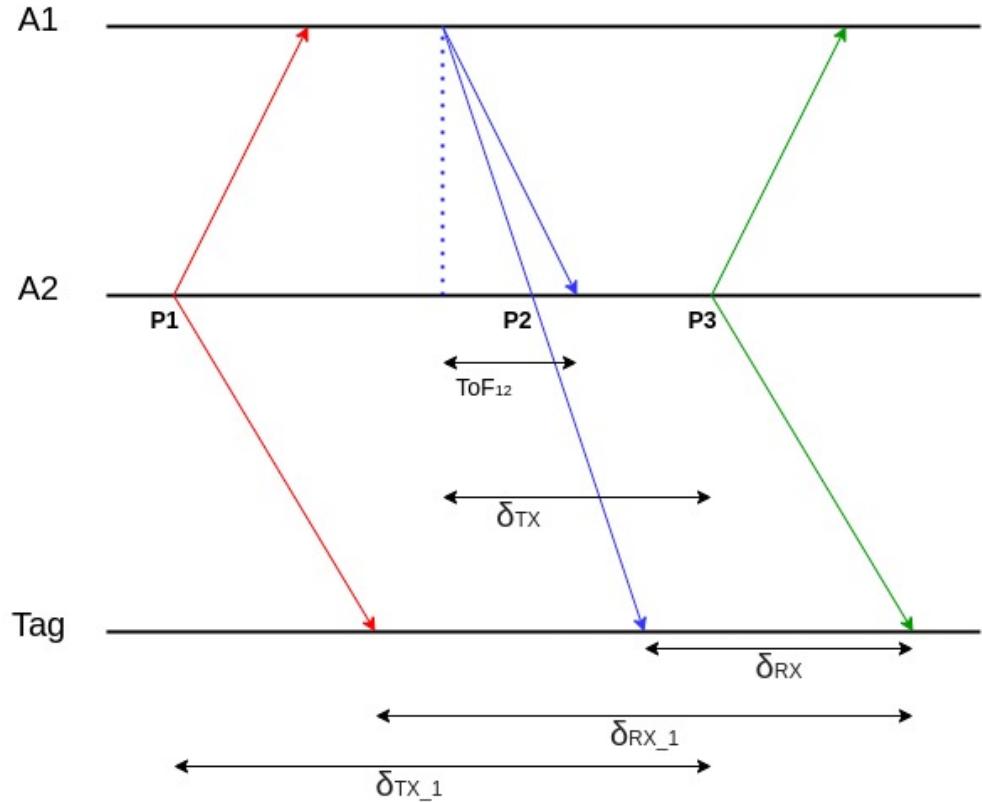


Figure 4.3: Diagram of the scheme for calculating the TDoA at the tag between two anchors. Here, a packet P1 is sent from A2 that is picked up by A1. A1 updates its recorded timestamps and ToF information before sending out a new packet P2. This is picked up by A2, that performs the same update and sends out a new packet P3. All three packets are picked up by the tag, and the respective timestamps are used to calculate the values necessary to determine the TDoA of this system.

contains general information on this packet, including its sequence number, transmission timestamp (recorded by A3's clock) and receive timestamp (recorded by the tag's clock). The remaining elements in Figure 4.2 not mentioned are administrative such as the *to* address that is always 255, indicating the packet is a broadcast packet. These elements are not essential for the following calculations and are ignored.

The important question now is: how does the tag calculate a time-difference of arrival value? Referring back to the system described in Figure 4.3, the tag has received all three packets complete with their timestamp information. However, the issue of clock synchronization resurfaces: these timestamps are generated by the clocks of any of these three nodes, which operate on the same frequency but are not synchronized. Simple TX/RX timestamp comparisons, then, will not be sufficient. Instead, a couple algebraic techniques are employed to bypass this.

The tag records the timestamps at which it received  $P_2$  and  $P_3$  from A1 and A2 respectively ( $T_{P2\_RX}$ ,  $T_{P3\_RX}$ ). The difference between when these signals were received can be denoted as:

$$\delta_{RX} = T_{P3\_RX} - T_{P2\_RX} \quad (4.1)$$

Since these are both recorded by the tag this difference can be calculated directly for an accurate value. Similarly, the difference between when these signals were sent can be described as:

$$\delta_{TX} = T_{P3\_TX} - T_{P2\_TX} \quad (4.2)$$

Here  $T_{P3\_TX}$  is the transmission timestamp for  $P_3$ , recorded at the transmitter A2 and sent as part of the packet.  $T_{P2\_TX}$  is similarly recorded at A1 and sent as part of the  $P_2$  packet.

Given the two timestamps in eq.4.2 are recorded by two separate clocks calculating  $\delta_{TX}$  requires a bit more legwork than  $\delta_{RX}$ . Consider the additional tools available to the system: not only do the packets contain their TX timestamps, but RX timestamps of other packets the transmitting anchor recently received from other remote anchors in the system. In addition, these packets contain subsequent time of flight calculations. Now,  $P_2$  is transmitted from A1 and picked up by the tag, *but it is also received and processed by A2* prior to  $P_3$ 's transmission. The timestamp at which A2 receives  $P_2$  is recorded ( $T_{P2\_RX@A2}$ ) and pushed as part of  $P_3$ 's remote anchor data in addition to the latest time of flight calculation between A2 and A1 ( $ToF_{12}$ ).

Recall from equation 3.7 the time of flight is quantified as the number of clock ticks observed for a signal to travel from one node to another. If the tag is passed the timestamp for  $P_2$ 's reception at A2 as well as the time of flight between A2 and A1 in clock ticks, it can be reasonably assumed that:

$$T_{P2\_TX} \equiv T_{P2\_RX@A2} - ToF_{12} \quad (4.3)$$

Equation 4.2 can now be modified to:

$$\delta_{TX} = T_{P3\_TX} - (T_{P2\_RX@A2} - ToF_{12}) \quad (4.4)$$

Now, both timestamps for calculating  $\delta_{TX}$  use only one clock as a reference.

The time-difference of arrival can now be calculated. Two signals, one from each anchor, are sent to the tag with a transmission time difference of  $\delta_{TX}$ . These signals are received at the tag with a reception time difference of  $\delta_{RX}$ . The TDoA, then:

$$TDoA = \delta_{RX} - \delta_{TX} \quad (4.5)$$

However, this isn't quite the complete picture yet. Recall no two clocks are perfectly identical: even precision clocks manufactured for the same frequency will inevitably feature imperfections in the clock crystal that cause frequency fluctuations. This will desynchronize two otherwise-identical clocks over time (and rather quickly for clocks operating on the nanosecond scale). This is referred to as *clock drift*, and to compensate for this the tag can use two consecutive packets from an anchor (see A2 in Figure 4.3) to keep track of that anchor's *clock drift coefficient* ( $\alpha$ ) with the tag's clock. To calculate  $\alpha$  the tag can compare the respective differences between reception and transmission timestamps of packets P1 and P3 respectively. Let these differences be denoted as  $\delta_{RX1}$  and  $\delta_{TX1}$ . Then:

$$\delta_{RX1} = T_{P3\_RX} - T_{P1\_RX} \quad (4.6)$$

and

$$\delta_{TX1} = T_{P3\_TX} - T_{P1\_TX} \quad (4.7)$$

Both  $P_3$  and  $P_1$ 's transmission timestamps are recorded at A2, and their reception timestamps are recorded at the tag. The tag's position relative to the anchor generally does not change between these packet transmissions (this is obviously not true for a moving tag, however the system updates with sufficient speed of hundreds of packets per second that such movement is usually negligible) meaning the differences between these values should ideally be identical. What will cause these values to deviate, then, comes down to the aforementioned frequency disparity between the two nodes' clocks. To single out this clock drift:

$$\alpha = \frac{\delta_{RX1}}{\delta_{TX1}} \quad (4.8)$$

This clock drift coefficient is measured with respect to A2, the same node at which the remote TX timestamps for equation 4.5 are recorded. This enables the tag to compensate for any potential clock drift in its tdoa calculations. The final tdoa calculation, then:

$$TDoA = \delta_{RX} - (\alpha * \delta_{TX}) \quad (4.9)$$

### 4.2.2 Noise and Outlier Filtering

At this point the tag is able to rapidly collect packets and update TDoA values between itself and various pairs of anchors reliably. However, there are two considerations to have before considering the localization algorithm: noise and outliers.

In regards to the former noise is a phenomenon in wireless communication where many elements such as outside electromagnetic interference cause interference and delays in the transmission of a signal and its contents. A facet of this experiment is in modeling the noise of the system rather than attempting to filter it (that step comes in incremental experiments that build on this thesis), and as such this noise can be observed as Gaussian variations around a median value. This noise also affects the Z-axis of the position estimation later in this section – further discussion and the noise model can be observed in the Results section of this document.

Outliers are caused by similar elements to noise in addition to other phenomena such as multipathing: these are generated values that fall considerably outside the expected parameters for measured values (for example: a series of ToF values around 1.1 meters followed by a value at 1093.23 meters). An accurate localization system is less able to ignore outliers as these can potentially "throw off" a position measurement, and these must be accounted for.

For this system accomplishing outlier filtering is very straightforward: for every new TDoA measurement between a given anchor pair and the tag the *qth percentile* of that value and the previous ten values for that pair is calculated. More specifically, if

the new TDoA value falls outside of the 5% to 95% mark set by the previous ten values it is considered an outlier and discarded. This allows flexibility for the tag to change its estimated position without the risk of the set of ten values converging into a tight set of values and discarding all future TDoA calculations as outliers. This method isn't perfect and better methods exist, however for the purpose of this thesis utilizing qth percentile filtering was sufficient.

#### 4.2.3 Hyperbolic Localization: Nonlinear Least Squares

The TDoA values are continuously updated. Noise is appropriately being modeled in a static system and outliers are appropriately filtered. The system is ripe for an implementation of a localization algorithm. Or, more specifically, a nonlinear estimator: here this is a method to estimate an appropriate 3D position (x,y,z) the hyperbolae generated by the TDoA values converge at.

Recall equation 3.12, or the TDoA of a 2D system:

$$c\Delta t = \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2} - \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2} \quad (3.12)$$

To reiterate the right-hand side represents a 2D hyperbola. Similarly, for 3D estimation, this expands to:

$$c\Delta t = \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2 + (z_t - z_1)^2} - \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2 + (z_t - z_2)^2} \quad (4.10)$$

The right-hand side of equation 4.10 represents a 3D *hyperboloid*.

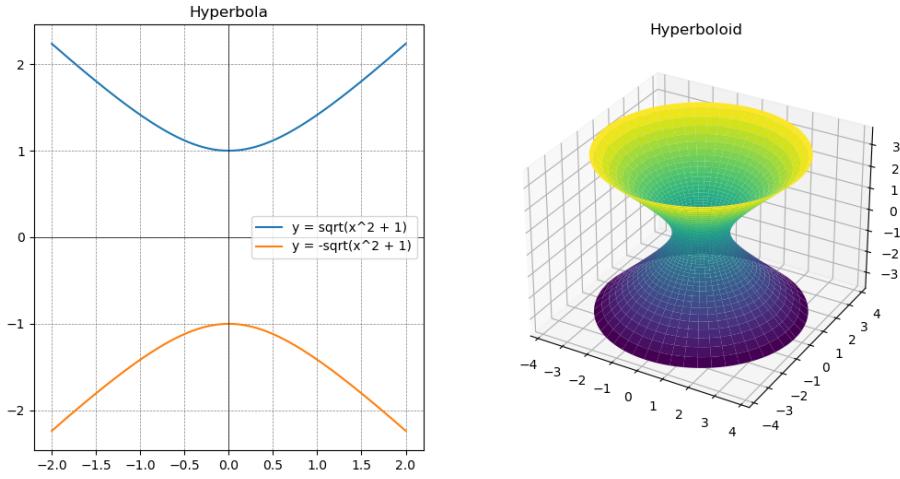


Figure 4.4: 2D Hyperbola vs 3D Hyperboloid

Just as with the hyperbola, the focii of the hyperboloid are the two anchors of the TDoA measurements and the surface of the hyperboloid depicts the possible 3D locations of the tag. With multiple TDoA pairs multiple hyperboloids can be generated with their intersecting surfaces reducing the total possible locations of the tag. Ideally, with four anchors and six generated TDoA pairs a sufficient number of hyperboloids are generated to create a singular intersection.

To locate this intersection mathematically the *nonlinear least squares* estimation method is employed. Given the tag can theoretically move in any direction in 3D space including forward and backward, this system is considered *nonlinear* since the dependent variable of the system (the tag's location) is not affected by the independent variable (time). The dynamic nature of the tag's position necessitates the use of a more advanced optimization technique – nonlinear least squares is particularly suited

for this since (true to its namesake) it can handle the intricacies of non-linear relationships effectively. By iteratively minimizing the sum of the squared differences between the observed and predicted TDoA values this method is able to accurately estimate the position of the tag. Mathematically, this is expressed as:

$$\sum_{i=1}^m r_i^2 = \sum_{i=1}^m (y_i - f(x_i, \beta))^2 \quad (4.11)$$

where  $r_i$  is the *residual*, or difference between observed data and model predictions, for the  $i$ -th observation,  $y_i$  represents the observed value,  $f(x_i, \beta)$  is the nonlinear model's predicted value given the currently-estimated location parameters  $\beta$ , and  $m$  is the number of data points considered.

The goal of this calculation is to begin with an initial guess for the tag's current position, sourced either from an initial setup value or from a previous position estimate, and iteratively update this guess to reduce the residual as much as possible. An important component to this optimization process hinted at here is the Jacobian matrix, which contains the partial derivatives (or rates of change with respect to certain elements) of all residuals in the system. This is expressed as:

$$J_{ij} = \frac{\partial x_i}{\partial \beta_j} \quad (4.12)$$

This helps us to understand how small changes in the position of the tag in 3D space affect the residuals, and each element of this matrix will tell you how much a small change in one coordinate will change a specific residual (more specifically, how

much will  $\frac{\partial}{\partial x}$  affect  $r_x$ ). Referring back, then, to equation 4.10 the system tracks a series of euclidean distances between the estimated position of the tag (represented by the subscript  $t$ ) and base station  $i$  (either 1 or 2 in the equation). Let's reduce this equation to better simplify later calculations:

$$\begin{aligned} c\Delta t &= \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2 + (z_t - z_1)^2} - \sqrt{(x_t - x_2)^2 + (y_t - y_2)^2 + (z_t - z_2)^2} \\ &= ||P_t - P_i|| - ||P_t - P_j|| \end{aligned} \quad (4.13)$$

where  $P_t$  is the tag's 3D position and  $P_i, P_j$  are the 3D positions of tags  $i, j$ . We can restate this as:

$$r_k = (||P_t - P_i|| - ||P_t - P_j||) - c\Delta t \quad (4.14)$$

where  $r_k$  denotes the residuals of the system: again these are the differences between the observed TDoA measurements and the estimated TDoA values based on the current tag position estimate  $P_t$ . Consider the partial derivative of the residual  $r_k$  with respect to the x coordinate of the system,  $P_x$ . Note the TDoA value ( $c\Delta t$ ) is a constant, therefore its derivative is zero and is filtered from this expression:

$$\frac{\partial r_k}{\partial P_x} = \frac{\partial}{\partial P_x} (||P_t - P_i|| - ||P_t - P_j||) \quad (4.15)$$

After some modifications using the chain rule the total change in  $r_k$  when you move  $P_x$  is:

$$\frac{\partial r_k}{\partial P_x} = \frac{P_{tx} - P_{ix}}{\|P_t - P_i\|} - \frac{P_{tx} - P_jx}{\|P_t - P_j\|} \quad (4.16)$$

Taking a step back, this residual is fed into an optimization algorithm and gives the direction in which the algorithm should adjust the estimated position to reduce the residuals. This is often described as a method of *nonlinear regression* to find the model that best fits a set of data points.

This is employed in the "least\_squares" function from the popular Python library "SciPy" [36]. This tool is specifically designed for solving non-linear least squares problems and performs a great deal of the aforementioned heavy lifting. Once the residuals are calculated for a given iteration this function utilizes the *Trust Region Reflective* algorithm, which adjusts its "trust region" or step size between iterations based on how well the model fits the data. To avoid the risk of steering too far from the intent of this thesis and getting "too deep in the weeds" this discussion will restrain from diving into the math behind this method.

The resulting least squares implementation takes in the updated TDoA calculations as input, breaks them down into component parts (see equation 4.14) and performs iterative calculations to estimate a 3D position for the tag. The TDoA calculations and localization algorithm are joined as part of a Python script; this script and all other scripts and firmware used for this experiment is available at the GitHub link in the abstract of this document.

# Chapter 5

## Results

The final experiment involves the tag remaining in a stationary location and connected to a laptop computer running the Python script. The four anchors send modified TWR packets to one another that are picked up by the tag and parsed for relevant timestamp data. Using aforementioned arithmetic methods the TDoA for each anchor pair ( $TDoA_{12}, TDoA_{13}, TDoA_{14}, TDoA_{23}, TDoA_{24}, TDoA_{34}$ ) can be rapidly updated as packets are received. All TDoA measurements are recorded separately to model the noise characteristics offline, while outliers are filtered out of the standard measurement updates. New TDoA measurements are fed into the nonlinear least squares model to iteratively update the tag's position based on these new parameters and the tag's previously-updated position (or, an initial guess). The end result is a system that consistently updates the 3D position of the tag, demonstrating a proof-of-concept for accomplishing the localization step of navigation.

Beginning with the noise model of the TDoA values, the following tables shows

five sample values (in meters) from each TDoA pair during the same period of time during the experiment.

$TDoA_{12}$	$TDoA_{13}$	$TDoA_{14}$
1.2587382387403099	-0.36178370264456583	0.5595250997730576
1.2546815038272028	-0.3662068364813419	0.5636845671107311
1.247970803737974	-0.37406038083872367	0.5497759118269621
1.241109982836588	-0.35447438898288997	0.5634978223742748
1.2587172928859083	-0.3473235246972672	0.5247309755317551

$TDoA_{23}$	$TDoA_{24}$	$TDoA_{34}$
-1.615873268516503	-0.7016441150388387	0.9357183848006798
-1.6258756536699623	-0.7016441150388387	0.9289121633656774
-1.6320052493409958	-0.7072130015134044	0.9289121633656774
-1.6239156586277526	-0.7043901370353174	0.9180122098991077
-1.6094930004664962	-0.7208263207395001	0.9150882414737123

Some standout data in these tables are the repeated values in pairs  $TDoA_{24}$  and  $TDoA_{34}$ . Other repeated values are interspersed through the long list of TDoA values: these are generated through the outlier filter that, upon finding an outlier, will discard that value and return the old TDoA estimation as the new value instead. These values aside, by observation the TDoA values in these tables have a very tight spread with a general variance of only a few centimeters.

The following Figures 5.1 and 5.2 are generated from a sample of approximately 70,000 packets from the experiment. The Gaussian plots feature the measured variance experienced by each TDoA pair, while the histograms demonstrate the raw values received by each pair.

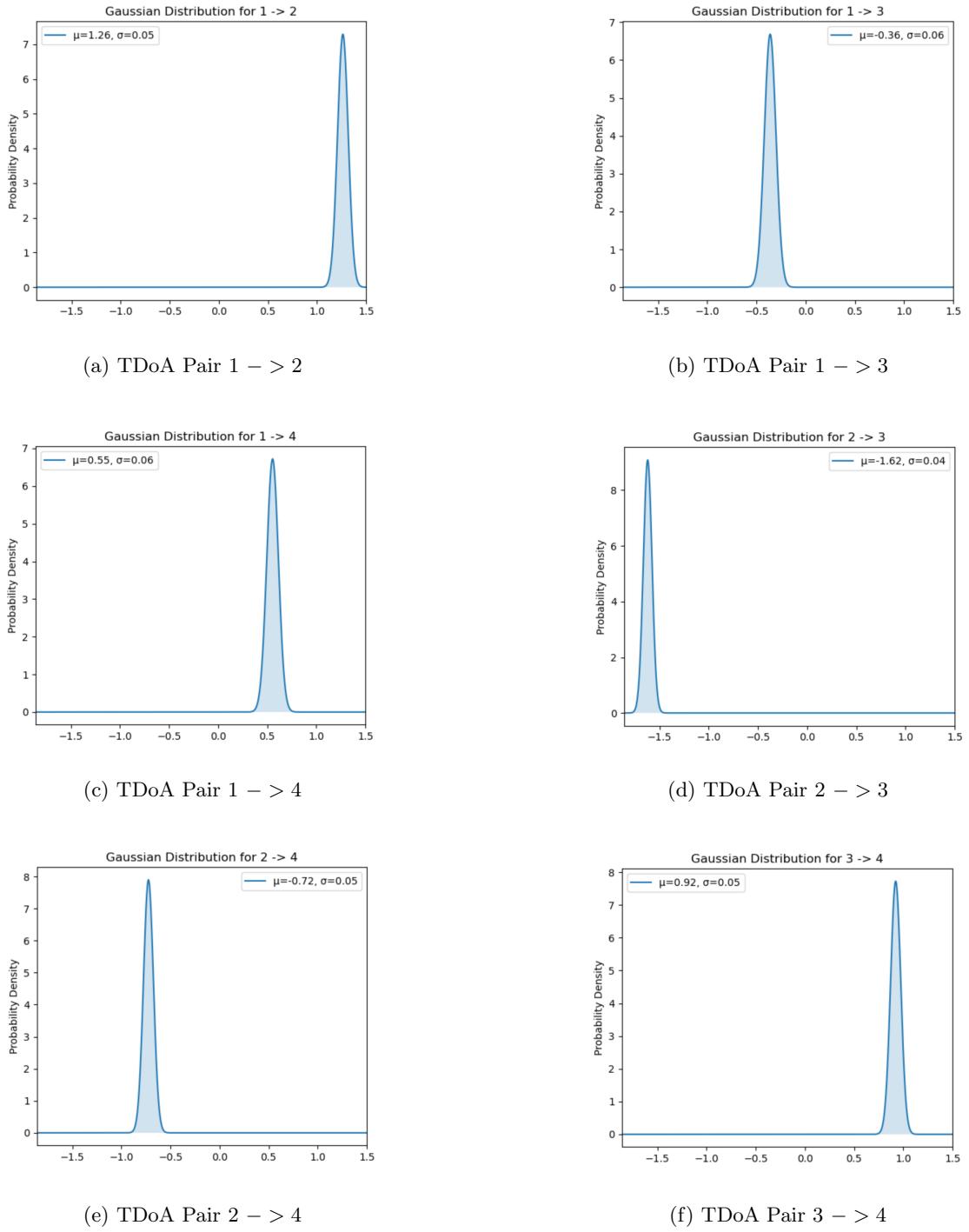


Figure 5.1: Gaussians generated by calculated TDoA values for each pair

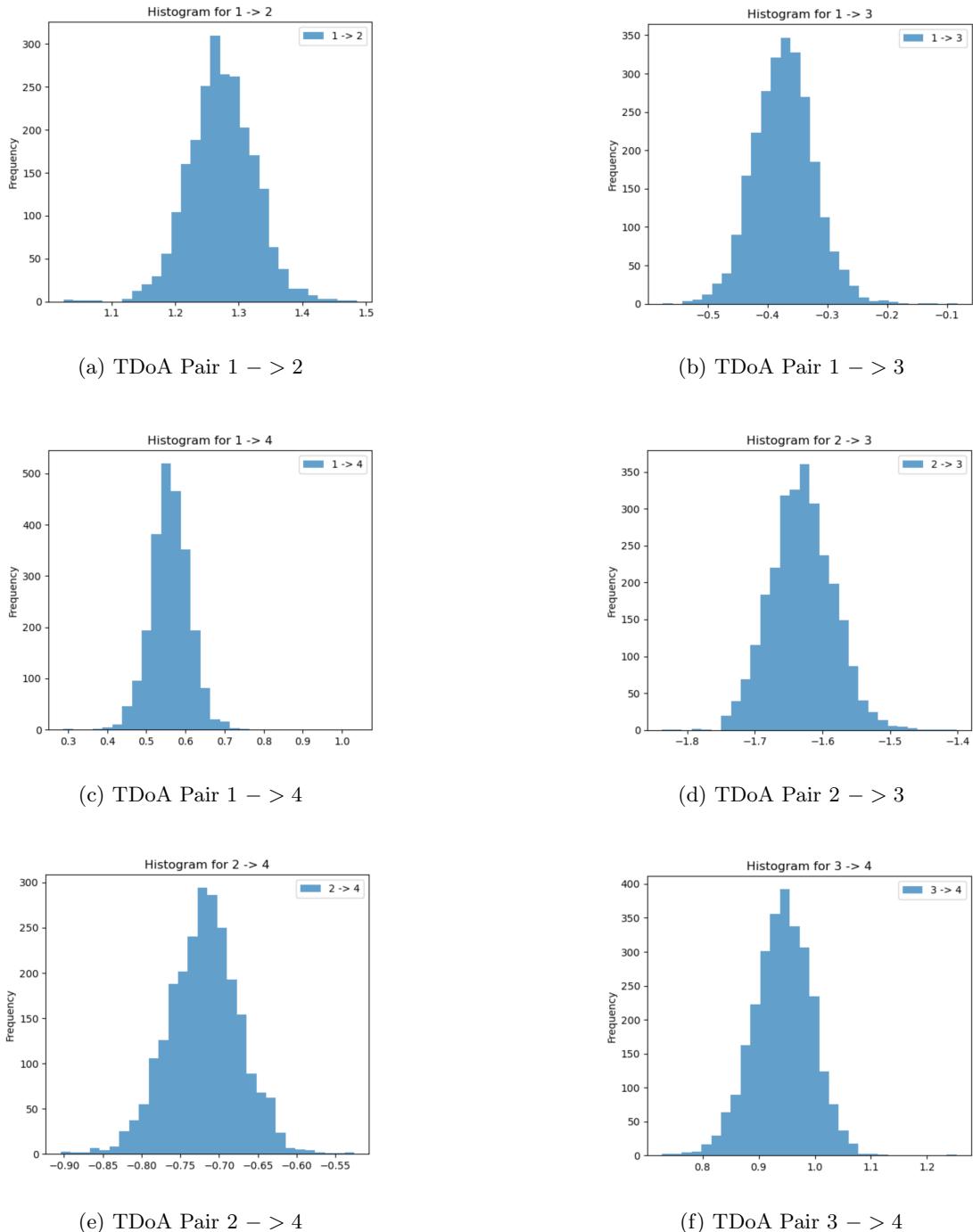


Figure 5.2: Histograms of TD**o**A values for each pair, generated from a sample of approximately 70,000 packets

The noise model reflects the small sample of values in the above tables: the bulk of values calculated for each TDoA pair fall within centimeters of the observed median value, displayed by both the peak of the pair's Gaussian and the tallest value in the pair's histogram. With a larger dataset of packets the histograms would be shaped ever closer to their Gaussian counterparts.

The outliers were sufficiently filtered out for these figures, however it's worth noting around 2 or 3 of every 100 packets returned an outlier, usually generated by some multipathing behavior extending the packet's travel time to its destination. While the qth percentile filtering method worked for this static test, an extraneous test where the tag was slowly moved around inside the convex hull resulted in several TDoA pairs *converging* on a tight range of values and rejecting all incoming TDoA calculations regardless if they were true outliers or not. This filtering method will need improvement before fielding on a moving system.

The localization algorithm also had some curious results. The following figures 5.3 and 5.4 depict the generated hyperboloids for a single point in time during the experiment, with half the TDoA pairs removed for better visibility of their intersection and estimation method.

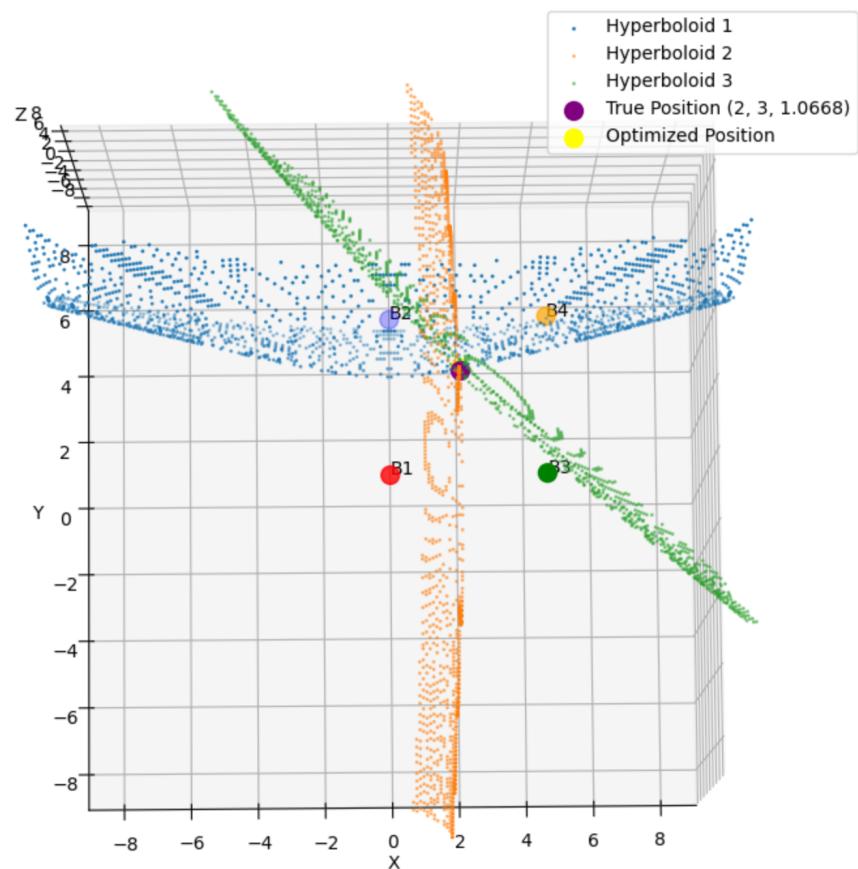


Figure 5.3: Top-view visualization of the nonlinear least squares position estimation to best show the curves of the hyperboloids and their intersect point

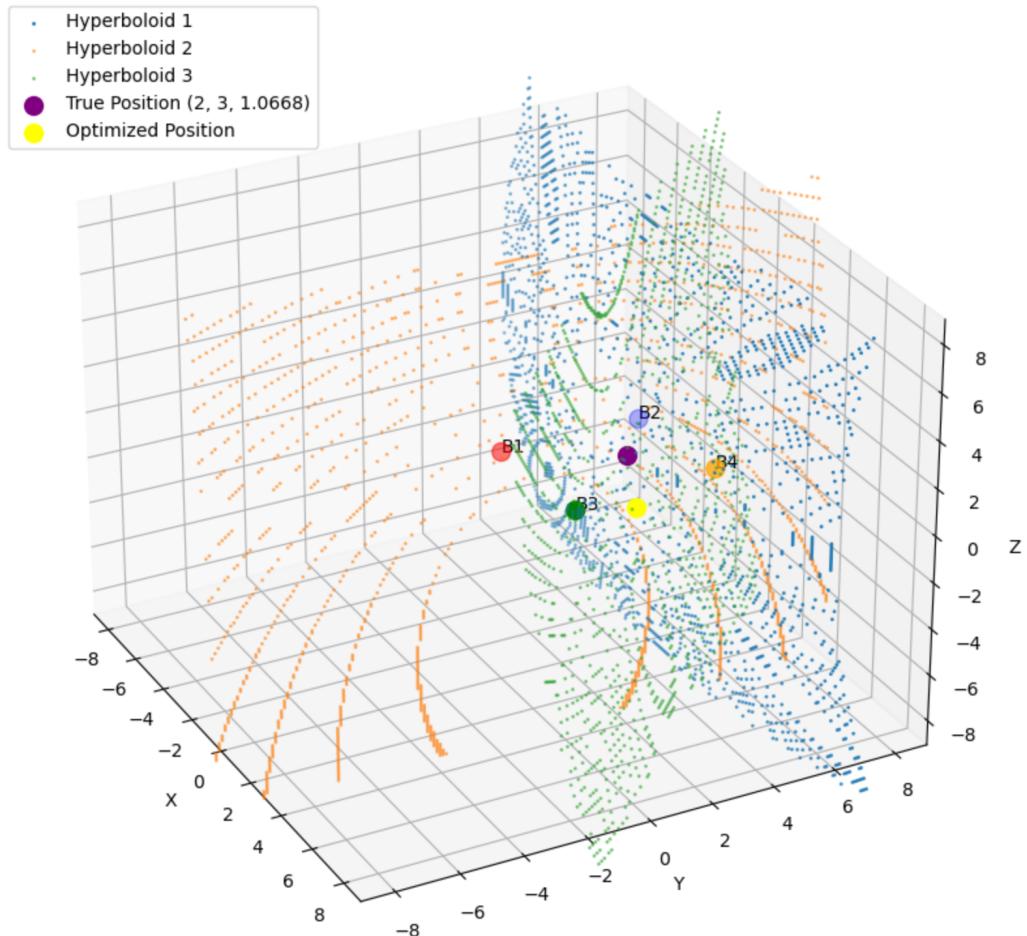


Figure 5.4: Side-view visualization of the nonlinear least squares position estimation.

Note the X and Y coordinates of the estimation (yellow dot) are extremely close to the true position (purple dot) while the Z coordinate has a considerable offset

As seen in Figure 5.4 the xy coordinates of the position estimate are remarkably close to the true xy coordinates of the tag. The z coordinate, however, suffers significant error and falling into the negative values (in essence: below the floor). Unfortunately the z coordinate also fluctuates considerably in practice, booting the possibility of a simple arithmetic or sign error. The cause of this is still an open-question at time of writing; that said the xy coordinates having this accuracy demonstrates a reliable proof-of-concept for an effective localization solution, requiring tuning rather than a rework. A solution to the z-axis drift will be one iteration of several in future work towards a full-scale, deployable solution.

# Chapter 6

## Conclusion

This thesis presented the development and evaluation of a Reverse Time-Difference of Arrival (RTDoA) algorithm intended for future deployments for 3D localization on a mobile aerial device. This evaluation focused on characterizing the noise experienced by the system when calculating TDoA for each of the anchor pairs relative to the tag, and testing a nonlinear least squares implementation for 3D localization. A static test in which the mobile tag was placed in a static position collecting packets within the *convex hull* drawn by the anchors was conducted, and each TDoA packet for each pair was recorded with a sample of 70,000 packets plotted in the Results section of this thesis.

The TDoA values experienced noise characteristics that are *Gaussian* in nature, with the bulk of values approaching some median value for each TDoA pair. The variance of these values was within centimeters of this median value. No more than 2% to 3% of the packets were filtered out as outliers due to multipathing or some other

errors; while the qth percentile filtering was successful it experienced debilitating convergence in a separate test where the tag was moved within the convex hull, and needs adjustment for future implementations involving a truly mobile tag. The nonlinear least squares implementation generated remarkably-accurate xy coordinate estimation, landing within centimeters of the true xy coordinates of the tag. However, the predicted z coordinate fell well outside acceptable parameters, consistently generating fluctuating and negative values (the tag was not below the floor for this test). This system sufficiently proves it is capable of generating accurate location estimations, but needs work to fix this z coordinate issue.

With future implementations, the computational efficiency and simplicity of this system make it very viable for RTDoA-based localization systems for applications such as autonomous navigation and indoor positioning. Future research with this system could explore integrating RTDoA with other localization techniques such as Terrain Relative Navigation (TRN) and Simultaneous Localization and Mapping (SLAM), serving as a reliable and accurate GPS-alternative to represent a ground truth for these systems. Overall, this work has laid a strong foundation for future advancements in RTDoA and precision navigation systems.

# References

- [1] What is Ultra-Wideband (UWB)? Here's everything you need to know | Bleesk, April 2020.
- [2] Prabin Kumar Panigrahi and Sukant Kishoro Bisoy. Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University - Computer and Information Sciences*, 34(8, Part B):6019–6039, September 2022.
- [3] Robert H. Rogne, Torleiv H. Bryne, Thor I. Fossen, and Tor A. Johansen. On the Usage of Low-Cost MEMS Sensors, Strapdown Inertial Navigation, and Nonlinear Estimation Techniques in Dynamic Positioning. *IEEE J. Oceanic Eng.*, 46(1):24–39, February 2020.
- [4] Gen Fukuda, Daisuke Hatta, Xiaoliang Guo, and Nobuaki Kubo. Performance Evaluation of IMU and DVL Integration in Marine Navigation. *Sensors*, 21(4):1056, February 2021.
- [5] Veerachai Malyavej, Warapon Kumkeaw, and Manop Aorpimai. Indoor robot localization by RSSI/IMU sensor fusion. In *2013 10th International Conference on*

*Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 15–17. IEEE.

- [6] Alvin Marquez, Brinda Tank, Sunil Kumar Meghani, Sabbir Ahmed, and Kemal Tepe. Accurate UWB and IMU based indoor localization for autonomous robots. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 2017–03. IEEE.
- [7] Tao Yang, You Li, Cheng Zhao, Dexin Yao, Guanyin Chen, Li Sun, Tomas Krajnik, and Zhi Yan. 3D ToF LiDAR in Mobile Robotics: A Review. *arXiv*, February 2022.
- [8] You Li and Javier Ibanez-Guzman. Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems. *IEEE Signal Process. Mag.*, 37(4):50–61, June 2020.
- [9] Adafruit Industries. Raspberry Pi Camera Module 3 Standard, May 2024.
- [10] Mohammad O. A. Aqel, Mohammad H. Marhaban, M. Iqbal Saripan, and Nap-siah Bt. Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5(1):1–26, December 2016.
- [11] Ignacio Parra Alonso, David Fernández Fernández Llorca, Miguel Gavilan, Sergio Álvarez Álvarez Pardo, Miguel Ángel Garcia-Garrido, Ljubo Vlacic, and M. Ángel Sotelo. Accurate Global Localization Using Visual Odometry and Digital Maps on Urban Environments. *IEEE Trans. Intell. Transp. Syst.*, 13(4):1535–1545, April 2012.

- [12] GPS.gov: GPS Accuracy, March 2022.
- [13] Frank van Diggelen and Per Enge. The World's first GPS MOOC and Worldwide Laboratory using Smartphones, September 2015.
- [14] Mevlut Yetkin Mustafa Berber, Aydin Ustun. Comparison of accuracy of GPS techniques. *Elsevier*, 2012.
- [15] D. Parsons. *The Mobile Radio Propagation Channel*. John Wiley & Sons, New York, NY and Toronto, Canada, 1992.
- [16] Marco Chiani and Andrea Giorgetti. Coexistence Between UWB and Narrow-Band Wireless Communication Systems. *Proc. IEEE*, 97:231–254, March 2009.
- [17] Wenda Zhao, Jacopo Panerati, and Angela P. Schoellig. Learning-based Bias Correction for Time Difference of Arrival Ultra-wideband Localization of Resource-constrained Mobile Robots, March 2021.
- [18] M. M. Saad, C. J. Bleakley, Michael Walsh, and Tingcong Ye. High accuracy location estimation for a Mobile Tag using one-way UWB signaling. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, October 2012.
- [19] Ankush Vashistha and Choi Look Law. A Novel E-DTDOA Based One-Way Ranging Using UWB-IR With Unsynchronized Anchors. *IEEE Transactions on Industrial Informatics*, 17(2):1274–1286, February 2021.

- [20] Jong-Hoon Youn and Yong Cho. *Performance Study of an Ultra-Wideband Indoor Localization and Asset Tracking System*. August 2010.
- [21] Pedro Reche-Soto, Daniel Rojas-Valverde, Alejandro Bastida-Castillo, Carlos D. Gómez-Carmona, Markel Rico-González, Luiz H. Palucci Vieira, Luca Paolo Ardigò, and José Pino-Ortega. Using Ultra-Wide Band to Analyze Soccer Performance through Load Indicators during a Full Season: A Comparison between Starters and Non-Starters. *Appl. Sci.*, 12(24):12675, December 2022.
- [22] IEEE Standards Association, June 2024.
- [23] Federal Communication Commission (FCC). Revision of Part 15 of the Commission's Rules Regarding Ultra WideBand Transmission Systems, 2002. Federal Register / Vol. 67, No. 95.
- [24] Fernando I. Rial, Henrique Lorenzo, Manuel Pereira, and Julia Armesto. Waveform Analysis of UWB GPR Antennas. *Sensors (Basel, Switzerland)*, 9(3):1454, 2009.
- [25] J. J. Perez-Solano, Santiago Felici-Castell, Antonio Soriano, and Jaume Segura García. Time synchronization enhancements in wireless networks with ultra wide band communications. *Comput. Commun.*, 186, January 2022.
- [26] Mickaël Delamare, Remi Boutteau, Xavier Savatier, and Nicolas Iriart. Static and Dynamic Evaluation of an UWB Localization System for Industrial Applications. *Sci*, 2(2):23, April 2020.
- [27] Yuan Zhou, Choi Look Law, Yong Liang Guan, and Francois Chin. Indoor Elliptical

Localization Based on Asynchronous UWB Range Measurement. *IEEE Trans. Instrum. Meas.*, 60(1):248–257, May 2010.

- [28] Moritz Ziegler, Amir Ehsan Kianfar, Tobias Hartmann, and Elisabeth Clausen. Development and Evaluation of a UWB-Based Indoor Positioning System for Underground Mine Environments. *Min. Metall. Explor.*, 40(4):1021–1040, August 2023.
- [29] Chunhua Han, Shunbiao Xue, Li Long, and Xiongquan Xiao. Research on Inertial Navigation and Environmental Correction Indoor Ultra-Wideband Ranging and Positioning Methods. *Sensors (Basel, Switzerland)*, 24(1), January 2024.
- [30] David Chiasson, Yuan Lin, Manon Kok, and Peter B. Shull. Asynchronous Hyperbolic UWB Source-Localization and Self-Localization for Indoor Tracking and Navigation. *IEEE IoT J.*, 10(13):11655–11668, February 2023.
- [31] DW1000 - Qorvo, June 2024. [Online].
- [32] David Pollard and Peter Radchenko. Nonlinear least-squares estimation. *J. Multivariate Anal.*, 97(2):548–562, February 2006.
- [33] Regina S. Burachik, Bethany I. Caldwell, and C. Yalçın Kaya. A generalized multi-variable Newton method. *Fixed Point Theory Algorithms Sci. Eng.*, 2021(1):1–31, December 2021.
- [34] Kun Zhao, Tiantian Zhao, Zhengqi Zheng, Chao Yu, Difeng Ma, Khaled Rabie, and Rupak Kharel. Optimization of Time Synchronization and Algorithms with

TDOA Based Indoor Positioning Technique for Internet of Things. *Sensors (Basel, Switzerland)*, 20(22), November 2020.

- [35] Git repository: [bitcraze/lps-node-firmware](https://github.com/bitcraze/lps-node-firmware), May 2024.
- [36] `scipy.optimize.least_squares` — SciPy v1.13.1 Manual, May 2024.
- [37] Anritsu. Time Difference of Arrival (TDOA) Application Note.
- [38] Faranak Nekoogar. *Ultra-Wideband Communications: Fundamentals and Applications*. Prentice Hall PTR, Upper Saddle River, NJ, 1 edition, 2006.