TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

DEVELOPMENT OF AN AUTONOMOUS UNMANNED AERIAL SYSTEM FOR RF SOURCE LOCALIZATION FOR STATIC TARGET

By:
Gautam Kumar Khadga (077BAS011)
Kunal Ray (077BAS018)
Madhusudhan Basnet (077BAS021)
Nikesh Ranabhat (077BAS025)
Roshan Sandilya Jha (077BAS035)

A PROJECT REPORT TO THE DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF AEROSPACE ENGINEERING

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING
PULCHOWK CAMPUS, LALITPUR, NEPAL

March 5 , 2025

COPYRIGHT

LETTER OF APPROVAL

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS PULCHOWK

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

LETTER OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "Development of an Autonomous Unmanned Aerial System for RF Source Localization for Static Target" submitted by Gautam Kumar Khadga, Kunal Ray, Madhusudhan Basnet, Nikesh Ranabhat and Roshan Sandilya Jha in partial fulfilment of the requirements for the Bachelor's Degree in Aerospace Engineering.

Supervisor: Arun Bikram Thapa

Assistant Professor

Department of Mechanical and Aerospace Engineering

Institute of Engineering, Pulchowk Campus

External Examiner: Er. Bishal Timilsina

CEO

Nepware Pvt. Ltd

Head of Department: Sudip Bhattarai (PhD) Assistant Professor

Department of Mechanical and Aerospace Engineering

Institute of Engineering, Pulchowk Campus

Date: 9th March, 2025

ABSTRACT

Localization is the process of estimating the position of a target node, which can be achieved through either range-based or range free methods.This project focuses on a range-based localization approach using the Received Signal Strength Indicator (RSSI), chosen for its simplicity, cost effectiveness, and practicality.The method is applied in Radio Frequency (RF) localization to determine the positions of static targets in indoor and outdoor environments.For static targets, RSSI is used to estimate distances, enabling localization through a multilateration technique.This method aims to minimize estimation errors and improve accuracy.In the indoor scenario, a 2.4 GHz WiFi router acts as the transmitter, while a 433 MHz omnidirectional antenna serves as the receiver.The path loss exponent, a key factor in modeling signal propagation, is calculated for both indoor and outdoor environments to assess the effects of distance and environmental factors on signal strength.The primary objective of this project is to maximize localization accuracy.In indoor environments,static sources are localized with minimal error.Additionally, the path loss exponent is computed for both short-range and long range outdoor scenarios to better understand how distance impacts signal propagation and localization precision.The project also addresses challenges such as noise, environmental factors, and hardware limitations, proposing solutions like filtering and antenna calibration to improve accuracy.Experimental results from several mission flights which effectively demonstrate the system's effectiveness and main goal of this project.The first mission flight achieved a distance error of approximately 6.16m,while the second mission flight reduced the error to 2.625m, similarly for the third and fourth mission flight achieved a distance error of approximately 1m and 70cm.Furthermore, analysis of the Pixhawk log file revealed a GPS error of 1m, which was accounted for to refine the accuracy of the static RF source localization.These results validate the system's capability to localize static targets with high precision,making it a practical solution for applications such as disaster rescue, environmental monitoring, and border surveillance.Future enhancements include dynamic target localization and obstacle avoidance system for more complex environments.

Keywords: RF Localization, RSSI, Multilateration, Path Loss Exponent, Signal Propagation, Indoor/Outdoor Localization.

ACKNOWLEDGMENT

We extend our sincerest gratitude to our project supervisor, **Asst.Prof.Arun Bikram Thapa (Department of Mechanical and Aerospace Engineering)**, for his consistent guidance, support, and inspiration throughout this major project.We also acknowledge Center for Energy Studies at the Institute of Engineering for providing us separate workspace necessary to conduct the experiment for indoor localization.

We are grateful to the Department of Mechanical and Aerospace Engineering, Institute of Engineering, Pulchowk Campus for granting us the necessary resources to conduct the project.We would like to express our thanks to Project Coordinator **Asst.Prof.Biman Rimal**.We would like to express our thanks to **Er.Nabin Bhandari** for his technical advice and assistance throughout the project.
A special note of thanks to **Er.Sandesh Parajuli, Er.Nirajan Gupta** and their team for their continuous support and for sharing invaluable insights from their project.We genuinely appreciate and deeply value any suggestions or feedback that could improve this project, and we wholeheartedly acknowledge them.Additionally,we extend our gratitude to **Aniket Gupt** for his assistance in the manual operation of the drone.

Any kind of suggestion or criticism will be highly appreciated and acknowledged.

Authors
Gautam Kumar Khadga (077BAS011)
Kunal Ray (077BAS018)
Madhusudhan Basnet (077BAS021)
Nikesh Ranabhat (077BAS025)
Roshan Sandilya Jha (077BAS035)

TABLE OF CONTENTS

List of Figures

List of Tables

LIST OF ACRONYMS AND ABBREVIATIONS

RF	Radio Frequency
GPS	Global Positioning System
RSSI	Received Signal Strength Indicator
AoA	Angle of Arrival
ToA	Time of Arrival
TDoA	Time Difference of Arrival
RSS	Received Signal Strength
HITL	Hardware in thre loop
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
Li-Po	Lithium Polymer
ESC	Electronic Speed Controller
APs	Access Points
WSN	Wireless Sensor Network
FOV	Field of View
SWR	Standing Wave Ratio
HITL	Hardware in the loop

CHAPTER 1: INTRODUCTION

1.1. Background

Unmanned Aerial Vehicle (UAV) can play a crucial role in location based applications and remote sensing networks.Especially, using UAVs for Signal Source Searching and Localization (SSSL) has significant advantages over terrestrial based SSSL approaches due to the ease of capturing RF signals at higher altitudes and the autonomous 3D navigation capabilities of UAVs.For example, UAVs can be utilized in military surveillance and border patrol missions that require high accuracy of signal source positioning, or for localizing intentional or non-intentional jammer.An approximate but fast SSSL approach using UAVs can save lives during disaster or emergency situations.Global Positioning System (GPS) is widely used in location finding.Since GPS devices use a lot of power, access to GPS signals will be blocked in the event of a power outage.A mobile device is the most popular way to obtain location data via GPS, but its short battery life prevents it from operating for very long.This is where radio based localization comes in handy because the radio signal source can run on less power for several months.Since the person or wildlife we are following possesses a radio frequency transmitting device, we therefore concentrated on target localization utilizing radio frequency source localization.Using localization algorithms, this technology locates people and wildlife by providing them with a Radio Frequency (RF) source.The existing localization approaches include visual characteristics, RF time of arrival,angle of arrival, time difference of arrival, Doppler and direction of arrival, and Received Signal Strength Indicator (RSSI).RSSI-based techniques are the cheapest in outdoor conditions although for indoor conditions they perform not very well [1].Target localization has been explored and applied utilizing fixed reference nodes, and now we are going to apply the same methods on a Unmanned Aerial System (UAS).

1.2. Problem statement

Natural calamities like flood and landslides claim lives of countless people and property. Timely,The Government has been focusing on rescue of the victims in short period of time and finding the missing ones.Thus, a reliable method need to be setup for the operation without much dependence on WiFi or GPS signals where they won't be readily available.

1.3. Objectives

1.3.1. Primary Objective:

This project primarily aims to assemble and develop a UAV that will be autonomous and will localize target by multilateration approach.

1.3.2. Specific Objectives:

- To develop receiver and transmitter prototypes.
- To develop a MATLAB/Python program for localization of a static target.
- To assemble a drone and make it autonomous.
- Include the different hardware and sensors needed for target localization.
- To create a code for the static target's autonomous target localization technique.

1.4. System Requirements

Accurate localization of RF sources by using unmanned aerial vehicle (UAV) needs a proper and specialized hardware and software components.The repaired hardware and software are listed below

1.4.1. Hardware Requirement

- Lora Module and esp 8266 microcontroller.
- RF transmitter (With antenna) capable of transmitting
- RSSI signal with minimum loss of signal strength.
- Antenna-equipped RF receiver that can pick up signals from an RF source.
- A drone that can hover steadily and has GPS and altitude control capabilities.
- A flight computer or data processor to gather, process, and evaluate data (such as a Raspberry-Pi) the RF information. Data is transferred from the drone to the computer via a communication method (such as WiFi or telemetry).

- Pixhawk for the autopiloting of uav.

1.4.2. Software Requirements

- For the autonomous of UAV, MAVSDK and pymavlink libraries are installed.
- Software for controlling drone flying (PX4 ArduPilot).
- Arduino IDE microcontroller.
- Software for processing RF signals (such as Python with the necessary libraries).
- Geolocation algorithms (such multilateration) to pinpoint the RF source's location.
- Using mapping software, such as Google Maps, to show the RF source's location in relation to other locations.
- For the conversions of GPS coordinate to cartesian coordinate,the required algorithm are added.

Figure 1.1: Visualization of the Project's Goal

1.5. Scope of Project

- For a variety of reasons, potential targets might not have GPS access.In these situations, the locals and a helicopter identify the target, which is a costly and time-consuming procedure.

- Ecologists track the migration of animals by tagging them with radio beacons and monitoring their travels.
- Good Utilization in visualizing, studying and observing the surrounding environment.
- It can be used for border surveillance.
- It can also be used in disaster rescue operations, like localization of avalanche beacons.
- In isolated trekking areas, it can be helpful to locate lost and injured hikers.

CHAPTER 2: LITERATURE REVIEW

Time of Arrival Time of Arrival (ToA), Time Difference of Arrival Time Difference of Arrival (TDoA), Angle of Arrival (AoA), RSSI, and other metrics related to the received radio signals can all be measured in order to estimate location.Even while ToA and TDoA have been shown to produce location estimates that are more precise[1],their implementation necessitates expensive hardware and they are very susceptible to timing errors.One popular range based technique that is simple to use but vulnerable to multi-path and signal fading is RSSI.Given two AoAs and their locations,AoA,sometimes referred to as Direction of Arrival (DoA),is a range free technique that can be used to calculate the transmitter's two dimensional coordinates.But to get precise results, highly directed antennas or an antenna array are needed.RSSI based localization can be performed effectively using LoRa[2].Since RSSI-based localization does not require complicated hardware,it is the correct choice for RF target localization[3].A ground Radio Frequency (RF) emitter can be localized by collecting measures of the Received Signal Strength Indicator (RSSI) at different positions.For this ,a set of waypoints covering the area of interest must be defined[4].Selecting an area can be challenging because a wrong direction will cause the vehicle to move away from the objective's location and it will drain the UAV's battery,limiting the available search time[5].The multilateration performance may be enhanced by improving ranging accuracy by mitigating the impact of environment related and receiver errors[6].Distance or range measurement can be achieved by RSSI measurement[7].

Current research has concentrated on employing many UAVs at once to localize a target[8]. Nevertheless,it becomes advantageous to employ a single UAV because it is quite expensive. Techniques for outdoor localization based on RSSI have been developed for this purpose.It has been stated that an RSSI-method based on the clustering method combined with Singular Value Decomposition can achieve accuracy as low as 7m [9].There are various techniques used for RSSI approach including trilateration and multilateration method.Multilateration method is more accurate than trilateration [10]For signal source search and localization SSSL, UAV with predefined waypath is used.Linear least square Linear least square (LLS) based localization scheme is used for its relatively lower computational complexity[11].

Target localization using RSSI is challenging due to noise characteristics.RSSI measurements give lower accuracy due to variable attenuation (path loss) and fading effects with high variance[12].Another factor that affects the RSSI is the antenna orientation and thus can affect the calculated distance between two transceivers.However, in a practical scenario the RSSI is affected by different factors like physical distance, reflections of objects, environmental parameters,movement of objects or change in the environment,antenna position and polarization etc[13].The localization accuracy can be further improved when the outliers

in RSSI readings are omitted[2].Due to significant unmodeled noise,it is difficult to model radio wave propagation and correlate measured signal strength with distance.So use of filter is necessary[14].Various filters like particle filter,Kalman filter, Extended Kalman filter can be used to mitigate the noise issue.Although with particle filter only one UAV is required but it has high computational complexity[15].Extended Kalman filter provides more accurate data than Kalman filter with lower deviation[16] Localization systems can be used in disasters areas in unconstrained outdoor environments where sensors lack knowledge of environmental noise and uncertainty (rough terrain, lightning changes).Although GPS improve localization accuracy, even without GPS the system performed with similar accuracy and consistency[17].It has been found that increasing the number of receiver nodes can increase the accuracy of locating transmitter location.Also using multilateration,location of transmitter can be estimated easily.[10] Using multiple UAVs increases the accuracy of localization but it is costly and mostly not feasible.Localization can be performed using a single UAV effectively[9].Particle filter is used for dynamic target.for dynamic targets, greedy optimization is used [18].The advantages to integrate the extended Kalman filter model into time difference of arrival model are robust tracking capabilities and prediction ability of future position of unmanned aerial vehicle[19].Simple omnidirectional RSSI sensors paired with an extended Kalman filter as estimator, is a promising approach in autonomous cooperative localization[20].Dynamic RSSI filter that calculates the mean of certain RSSI values obtained from multilateration can improve the localization accuracy[21].

The preliminary aim of the project will be to localize a RF signal source using RSSI approach.We use multilateration algorithm for localization.The mission test flight will be carried on the open-controlled environment within the visual line of sight as limited by Nepalese Civil Airworthiness Requirements (NCAR) regulation.The application of vision sensors in the collaborative task of multiple UAVs can effectively avoid navigation interruption or precision deficiency caused by factors such as field-of-view obstruction or flight height limitation of a single UAV sensor and achieve large-area group positioning and navigation in complex environments[22].

Localization especially in urban environments can be challenging due to interference and noise.Strategy where UAVs autonomously plan their paths to locate RF sources makes the localization process effective[23].WiFi fingerprinting with RSSI can be used to localize a target in an indoor setting however,the fluctuation of wireless signals resulting from environmental uncertainties leads to considerable variations in RSSI[24].Software-in-the-loop system can be used to exhibit the conceptual proof with realistic models[25].Localization of moving objects in real time is a challenge,complex calculations are required.Critical parts of the real time localization process,for which calculations must be performed in a short period of time, such as triangulation[26].Autonomous UAV can be used to track a dynamic target using particle filter and a partially observable Markov decision process (POMDP) for dynamic path

planning[27].Recent advancements in drone surveillance systems highlight the importance of RF and WiFi-based target localization techniques for detecting and neutralizing potential threats, providing a critical foundation for the development of autonomous unmanned aerial systems for static RF source localization[28].WiFi based localization, utilizing both traditional RSSI ranging and the more recent FTM technique, offers a low cost and low complexity solution for real time UAV positioning, with FTM demonstrating a 37% improvement in accuracy over RSSI, making it a viable backup for GNSS denied environments in static and mobile UAV applications[29]. Done coordination in GPS denied areas are discussed, addressing issues associated with localization and coordinating multiple agents as they attempt to accomplish a common goal[30].We implement FALCON via a custom designed multi drone platform and demonstrate up to 2 times localization accuracy compared to a baseline flocking approach, while spending 37% less time localizing targets[31].Indoor localization (IL) systems are crucial for enhancing operational efficiency, safety, and user experience by allowing precise tracking of objects, robots, or individuals within various environments such as healthcare, retail, and industrial sectors[32].UWB radio has unique transmission characteristics which make picoseconds resolution timing possible, a requirement for centimeter accuracy in positioning applications[33].The sensors are purposely simple and cheap, meaning that conventional localization techniques,such as global navigation satellite systems are not feasible[34].UAVs can act as a data logger and enable localization in such an integrated environment, that can help us develop mission critical applications which can be deployed in harsh environments[35].Localization is an essential tool that ensures preparedness, response, and coordination between first responders and impacted people[36].Particularly, the received signal strength indicator is adopted as an auxiliary measuring component[37].

Autonomous aerial robots provide new possibilities to study the habitats and behaviors of endangered species through the efficient gathering of location information at temporal and spatial angularities not possible with traditional manual survey methods.A novel autonomous aerial vehicle system Tracker Bots to track and localize multiple radio tagged animals was presented.The simplicity of measuring the received signal strength indicator (RSSI) values of very high frequency (VHF) radio collars commonly used in the field is exploited to realize a low cost and lightweight tracking platform suitable for integration with unmanned aerial vehicles (UAVs)[38].Unmanned aerial vehicles (UAVs) are considered one of the most promising emerging technologies to support rescue teams in disaster management and relief operations according to UN and Red Cross reports.In this work, we consider a disaster scene with damaged communication infrastructure and leverage UAVs for efficient and accurate positioning of potential survivors through the seamless collection of the received signal strength indicators (RSSI) of their mobile devices[39].The field of indoor localization is fast devel-

oping and has important ramifications for a number of areas, such as smart infrastructure development, healthcare settings, industrial automation, and military operations[40].

CHAPTER 3: METHODOLOGY

The following flowchart will display the project's methodology.

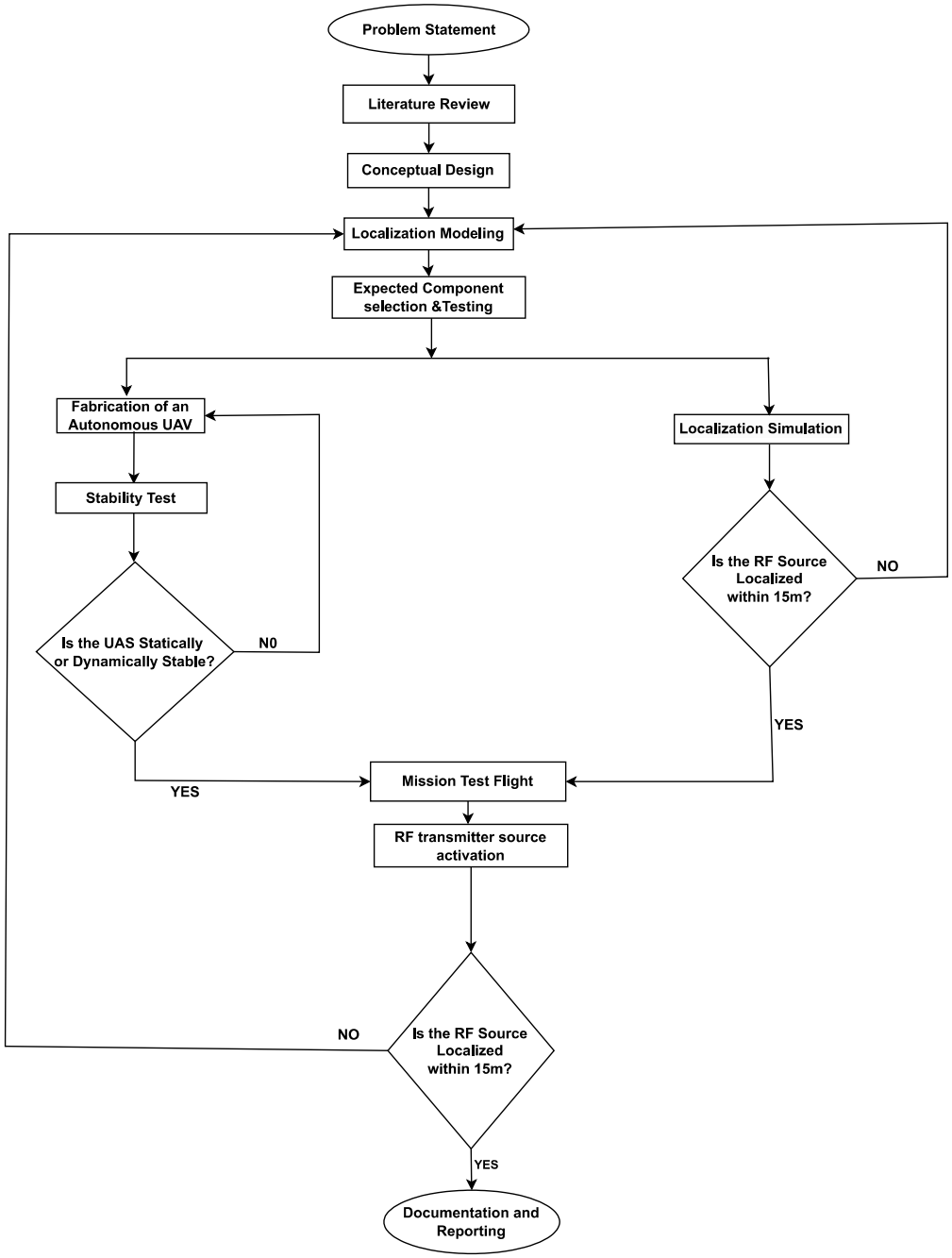

Figure 3.1: Flowchart of Methodology

3.1. Modelling Localization

Our work paper's primary goal is to employ autonomous UAS to locate a stationary RF transmitter with the highest precision achievable.We assume that drone has perfect knowledge of its location through use of GPS.Since the drone is limited to a fixed altitude, altitude is not included in the drone state.Changes in drone altitude do not affect the gain of most radio localization antennas,which have a nearly constant gain throughout elevation angle to the radio source.We,here use autonomous UAV discarding and removing the faulty readings thereby considering the best ones.The environmental factors like humidity,temperature,noises and other disturbances significantly affect the accuracy.In order to improve accuracy and reduce the noise we shall use Simple moving average(SMA) filter,Extended Kalman filter as per the need in our RSSI data.We take the thing into consideration that the position of the drone is known by GPS module with which the RSSI is calculated.Two processes can be performed for localizing target source.We can use either RSSI with trilateration or RSSI with Multilateration.For accuracy we use multilateration approach.

3.1.1. Distance estimation model

RSSI measurements, which are given in decibels (dBm), show the strength of the received signals as they move from transmitter to receiver.Due, to the fact that signal intensity diminishes with distance, RSSI data can be used to estimate the distance between the transmitter and receiver.One of the most widely used models for estimating distance in radio wave propagation is the log-normal model, which is supplied by

$$RSSI = RSSI_0 - 10n\log_{10}\left(\frac{d}{d_0}\right) - X(\sigma) \tag{3.1}$$

The effect of $X(\sigma)$ can eliminated to some extent by filtering.If we neglect $X(\sigma)$, and take the reference distance d_0 equal to 1 meter, then equation 3.1 becomes:

$$RSSI = RSSI_0 - 10n\log_{10}(d) \tag{3.2}$$

Where $RSSI_0$ can be experimentally determined, and the value of the path loss exponent n can be calculated by minimizing the sum of squares.The function to minimize is:

$$f = \sum (RSSI - RSSI_0 + 10n\log_{10}(d))^2 \tag{3.3}$$

The function is to minimize in this case is f. Then, it is possible to approximate the distance d between the transmitter and the receiver as:

$$d = 10^{\frac{RSSI_0 - RSSI}{10n}} \tag{3.4}$$

3.1.2. Simple RSSI

RSSI (Received Signal Strength Indicator) is a method to determine the Radio Frequency RF source based on the signals received by the receiver from the transmitter.This is the easiest and simplest way to perform localization due to lesser number of hardware and software requirement.Thus, we have to consider a lot of signal loss, attenuation, environmental barriers like humidity, temperature,etc.With the increase in distance between transmitter and receiver the signal strength gets weaker.There are algorithms used to convert the received signals into the corresponding distances.Received RSSI values is translated to the distance value which helps predict the target node P(x, y).

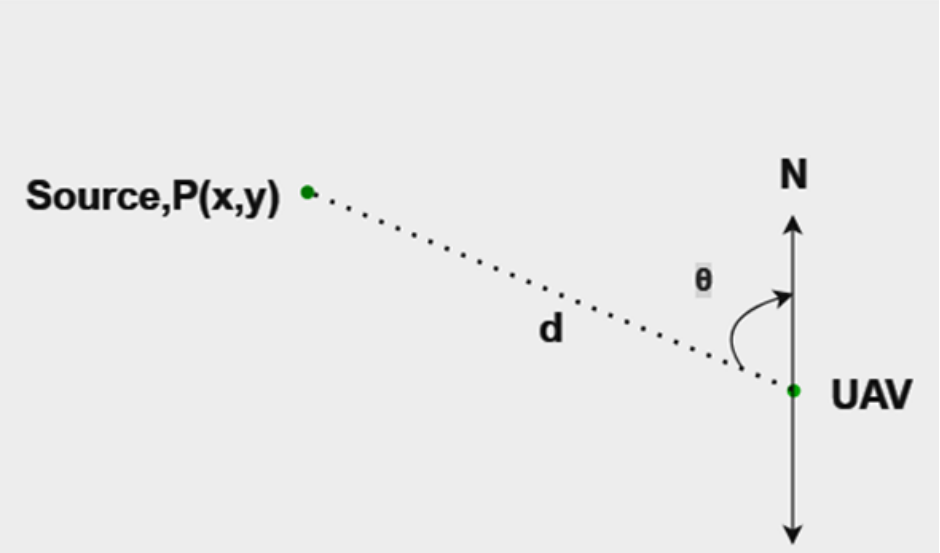

Figure 3.2: Simple RSSI-based localization model

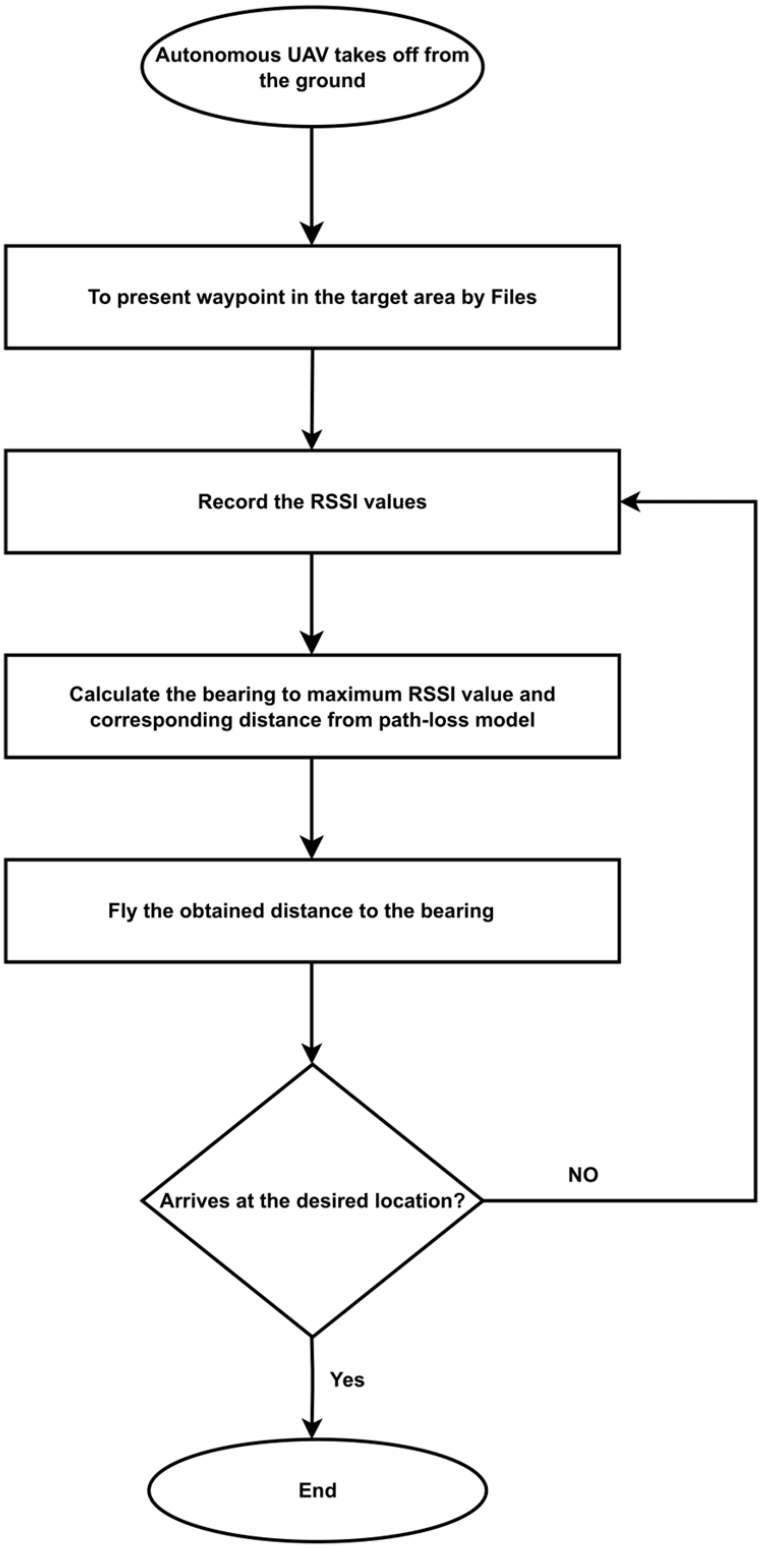

Figure 3.3: Simple RSSI Localization flowchart

3.1.3. RSSI with Multilateration

This method involves use of signal strength received by receivers at multiple locations to determine the position of transmitter .The signal is transmitted continuously and RSSI provides the indication of signal.The distance between the transmitter and receiver is estimated using the RSSI measurements.The separation and signal strength are inversely proportional to each other.The multilateration algorithm are used to estimate the position of the transmitter based on signals received.Multilateration localization can be performed in real-time as the receiver moves within the coverage area of the beacons.Continuous RSSI measurements and distance estimations enable static tracking of the transmitter's position.This technique is quite powerful and good compared in terms of accuracy due to its easiness and flexibility.For multilateration substitute n=4 in the below equations:
The system of equations is as follows:

$$(x-x_1)^2+(y-y_1)^2=d_1^2 \tag{3.5}$$

$$(x-x_2)^2+(y-y_2)^2=d_2^2 \tag{3.6}$$

$$\vdots$$

$$(x-x_n)^2+(y-y_n)^2=d_n^2 \tag{3.7}$$

Equation (3.5),(3.6),(3.7) will be linearized as presented in Equations (3.8) to (3.9) by subtracting the last equation from the n-1 previous ones:

$$-2(x_1-x_n)x-2(y_1-y_n)y=(d_1^2-d_n^2)-(x_1^2-x_n^2)+(y_1^2-y_n^2) \tag{3.8}$$

$$-2(x_2-x_n)x-2(y_2-y_n)y=(d_2^2-d_n^2)-(x_2^2-x_n^2)+(y_2^2-y_n^2) \tag{3.9}$$

$$\vdots$$

$$-2(x_{n-1}-x_n)x-2(y_{n-1}-y_n)y=(d_{n-1}^2-d_n^2)-(x_{n-1}^2-x_n^2)+(y_{n-1}^2-y_n^2) \tag{3.10}$$

These equations can be rewritten in matrix form as:

$$AX=B \tag{3.11}$$

Where:

$$A=\begin{bmatrix} -2(x_1-x_n) & -2(y_1-y_n) \\ -2(x_2-x_n) & -2(y_2-y_n) \\ \vdots & \vdots \\ -2(x_{n-1}-x_n) & -2(y_{n-1}-y_n) \end{bmatrix}, \quad X=\begin{bmatrix} x \\ y \end{bmatrix}, \quad B=\begin{bmatrix} (d_1^2-d_n^2)-(x_1^2-x_n^2)+(y_1^2-y_n^2) \\ (d_2^2-d_n^2)-(x_2^2-x_n^2)+(y_2^2-y_n^2) \\ \vdots \\ (d_{n-1}^2-d_n^2)-(x_{n-1}^2-x_n^2)+(y_{n-1}^2-y_n^2) \end{bmatrix}$$

Finally, the estimated location coordinates of the transmitter can be solved using:

$$X=A^{-1}B \tag{3.12}$$

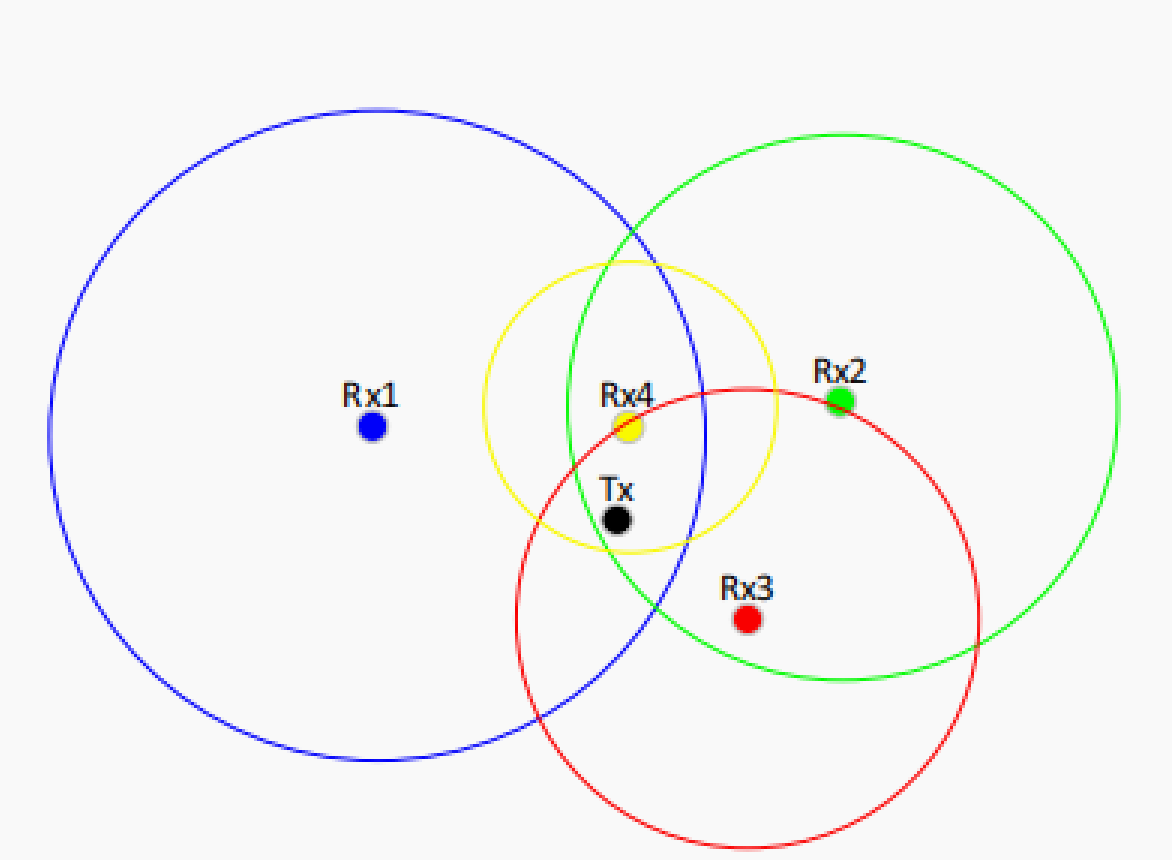

Figure 3.4: RSSI based Multilateration Localization Model

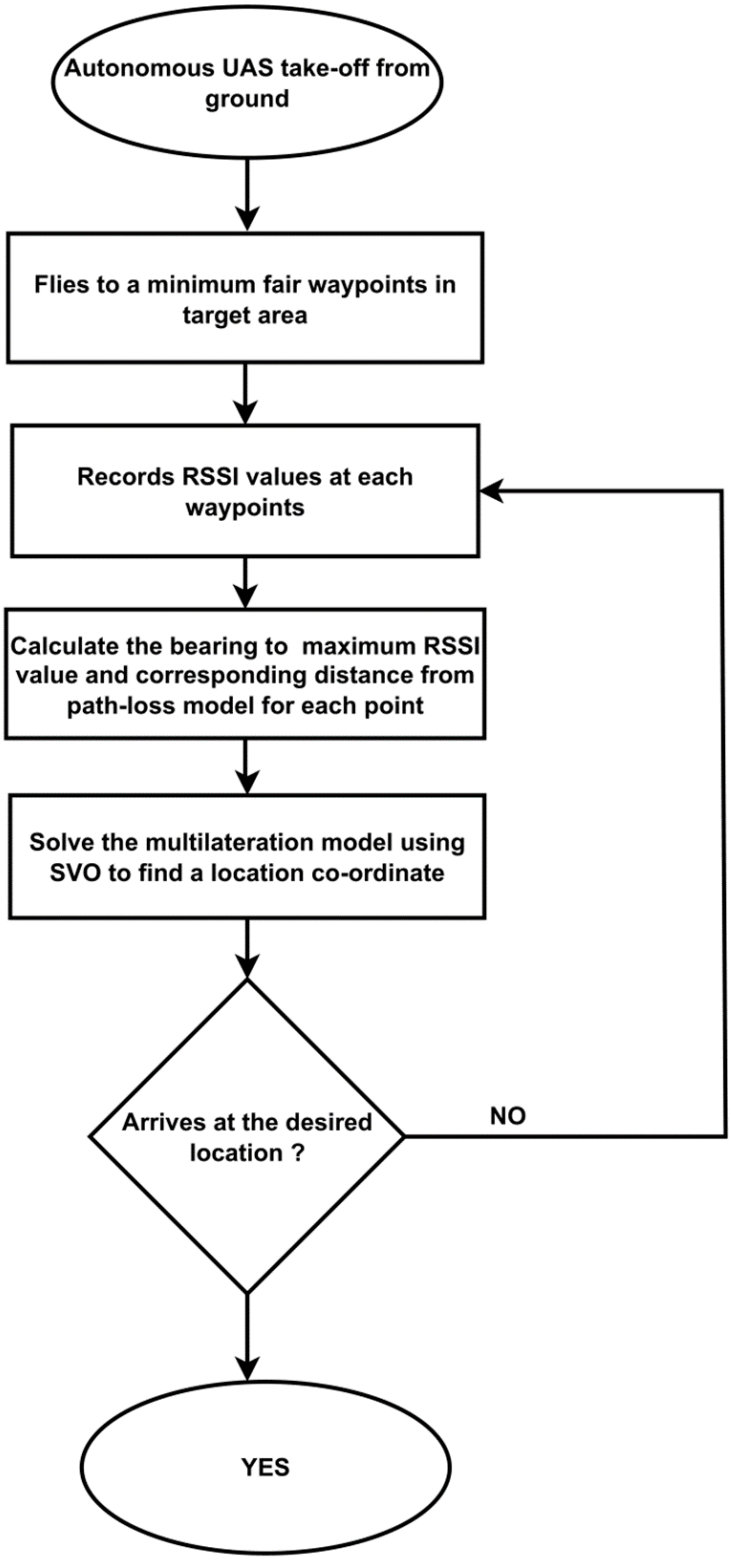

Figure 3.5: RSSI with Multilateration flowchart

3.2. Mission Flight Flowchart

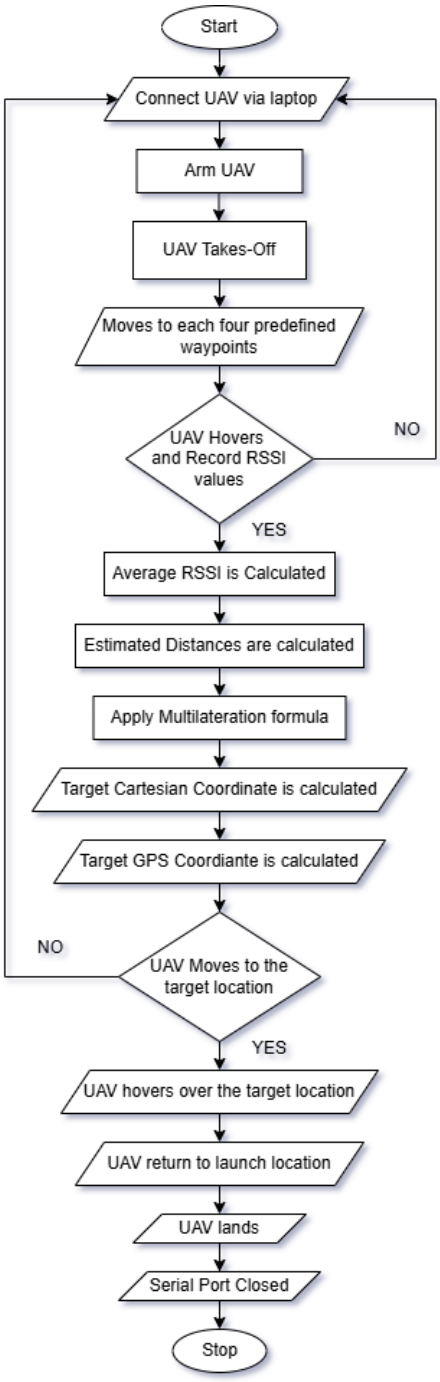

Figure 3.6: Mission flight flowchart

3.3. UAS Development

Four 1000 kv DC brushless motors make up the X-type drone layout that was utilized.The drone arrangement also has four electrical speed controllers to adjust the speed. A Pixhawk 4 flight controller was utilized to operate the drone, and a lighter and more robust drone frame was used.The flight computer will be a Raspberry Pi 4.The SX1278 LoRa transceiver ,and other peripheral parts and sensors will be included.A three cell Li-Po battery was utilized to power every component.Propellers is selected in accordance with the mission's specifications.Pixhawk can obtain the information it requires to oversee and finish the mission by connecting a GPS module and telemetry.The drone will be equipped with a variety of sensors to carry out the localization mission.Following drone assembly,tests was conducted to evaluate the drone's performance, control,and stability,and any necessary adjustments will be made.

Figure 3.7: Assembled UAS(Quadcopter)

Figure 3.8: Assembled UAS(Quadcopter)

The drone hardware components used for construction consist of:

1. Motor:Motors function as the essential components of a drone to drive propellers for generating required thrust.A drone requires identical count and clockwise rotation of motors to achieve balanced flight.Motor performance is determined by its KV rating that represents RPM production at each applied voltage in unloaded flight.Using high KV motors together with big propellers produces high speed RPM that requires excessive torque and results in overheating and motor failure.The drone incorporates four brushless motors having a 1000kv rating.

2. Propellers:Flight lift depends entirely on propellers since they serve to produce this necessary force. Motor size and pitch determine their operational output because pitch specifically represents the amount propellers travel during a complete revolution.The pitch elevation of a propeller determines its lifting capacity whereas the pitch reduction brings increased rotational torque.The weight-carrying capacity depends on using large propellers with reduced pitch but quick and precise manoeuvres require small propellers with raised pitch.A 10*4.5 (inch) propeller is attached to each motor.The

diametrically opposed propellers revolve similarly when two of them rotate in a clockwise direction and the other two in an anticlockwise direction.

3. Electronic Speed Controller (ESC) :The Electronic Speed Controller plays a crucial role within drones by controlling the precise speed of electric motors.The signals from the flight controller drive the ESC to adjust motor voltage which regulates the RPM for drone movement control.Speed Controller Electronic devices display their ability to modify motor speed through their refresh rate expressed in Hertz while their maximum current rating shows the steady current capability of the motors.An ESC rated at 10A can sustain a 10A continuous current delivery.The current rating of an ESC exceeds the maximum current demand of the motor to protect the system.The enclosure on ESCs shows their burst rating to indicate their temporary maximum current tolerance before damage occurs.A 30A ESC operates on the drone system to control motor speed regulation.

4. Battery:All electrical and electronic components of the drone run through the battery which operates with a purpose to provide power.The standard voltage for Lithium-Polymer (Li-Po) batteries amounts to 3.7V per cell in their design.Series connection of battery cells increases total voltage output while the battery rating reveals its number of interconnected cells through the "S" label in examples like the 3S configuration signifies three series cells. Determining the ideal battery demands an evaluation between battery capacity (measured in mAh) that determines power supply duration while the battery C-rating demonstrates its capability for rate of discharging.

Figure 3.9: 3 Cell LiPo Battery

5. Pixhawk 4:During the localization mission, the quadrotor is controlled by the primary flying controller, Pixhawk 4.The Pixhawk used in the quadrotor system is linked to the PM02 GPS module.

6. M8N GPS module:Often utilized in drone applications, the M8N GPS module is a high-performance GPS receiver.It gives precise GPS location information to the drone,

which is necessary for precise navigation and flight control.A built-in compass on the M8N GPS module helps with localization algorithms.Several GPS satellites send data to the M8N GPS module, which then calculates the position of the drone based on the information obtained.The drone's flight controller receives the GPS data and uses it for control and navigation. Because it makes it possible for the drone to operate safely and dependably, the M8N GPS module is an essential part of drone applications.

7. PM02:The PM02 Power Module is made to effectively transfer battery power to different onboard components in drones and other remote-controlled vehicles.It provides controlled power to the electronic speed controllers (ESCs), flight controller, and as well as other electronic gadgets.The PM02 makes wiring connections easier and produces a steady voltage output, which improves the drone's overall performance and dependability.It is a crucial part of preserving flight stability and system efficiency because of its small size and integrated current and voltage sensors, which guarantee steady power delivery.

8. Raspberry Pi 4:Integrating a Raspberry Pi 4 into an Unmanned Aerial Vehicle (UAV) project significantly enhances its computational capabilities, enabling advanced autonomous functions.Serving as a companion computer, the Raspberry Pi 4 processes complex algorithms for tasks such as real-time object detection, environmental mapping, and navigation.Its compatibility with various sensors and cameras allows for seamless integration, facilitating data acquisition and processing.Utilizing open-source flight controllers like Pixhawk,the Raspberry Pi 4 can execute high level mission planning and control through software such as DroneKit, which offers Python-based scripting for autonomous missions.This setup supports real-time video streaming and telemetry data transmission, essential for surveillance and reconnaissance applications. Moreover, the Raspberry Pi 4's connectivity options,including WiFi and potential 4G modules, enable extended-range operations and remote control capabilities.The platform's flexibility allows for the implementation of machine learning models, enhancing functionalities like precision landing and obstacle avoidance.Overall,the integration of a Raspberry Pi 4 into a UAV system provides a robust framework for developing sophisticated,autonomous aerial solutions.

3.4. LoRa-ESP8266

A transmitting device prototype was developed using an SX1278 LoRa module and an ESP8266 microprocessor.This module served as a receiver as well.The gadget starts generating RSSI packets on the 433 MHz frequency band, an Asia-specific RF band that is license-free.

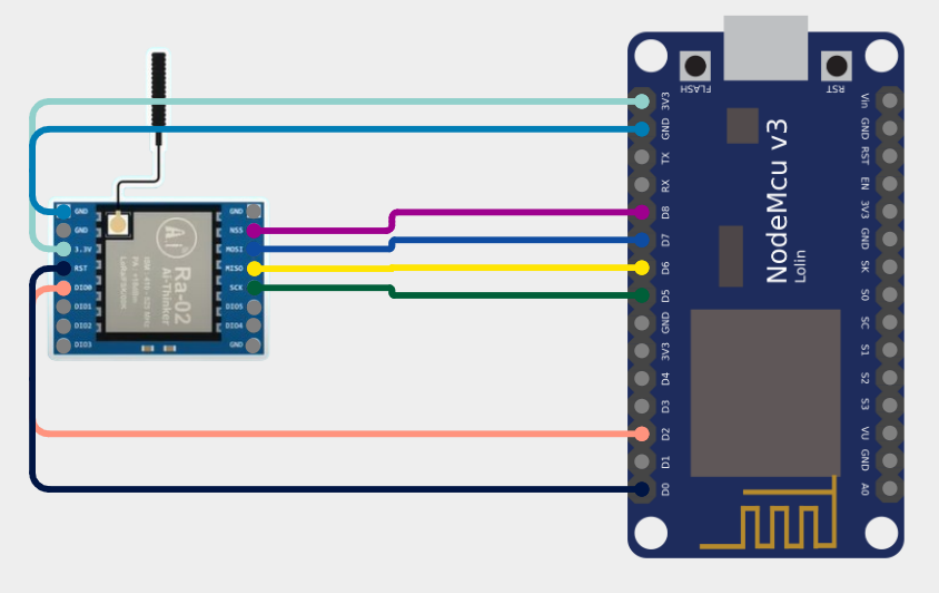

Figure 3.10: Schematic Diagram of LoRa Module and ESP8266

3.4.1. Prototype Transmitter

The prototype transmitter was developed using an SX1278 transceiver and a NodeMCU ESP8266 microcontroller.Its performance was validated through calibration in various locations.

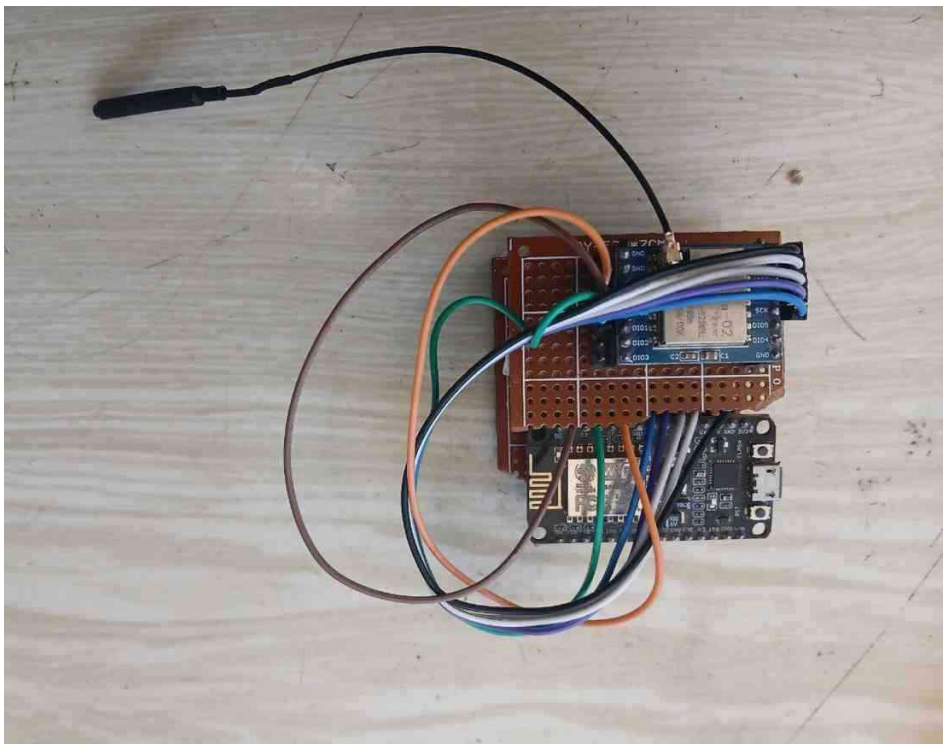

Figure 3.11: Prototype Transmitter

3.4.2. Schematic of Prototype Transmitter and Receiver

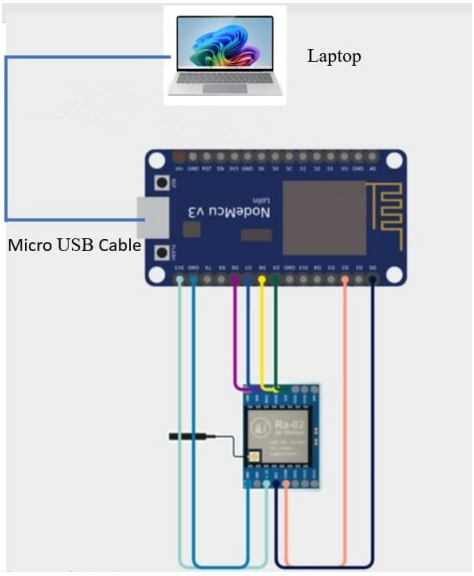

Figure 3.12: Schematic Diagram Prototype Transmitter and Receiver

3.4.3. Prototype Receiver

The prototype receiver was developed using an SX1278 transceiver and a NodeMCU ESP8266 microcontroller.Its performance was validated through calibration in various locations.

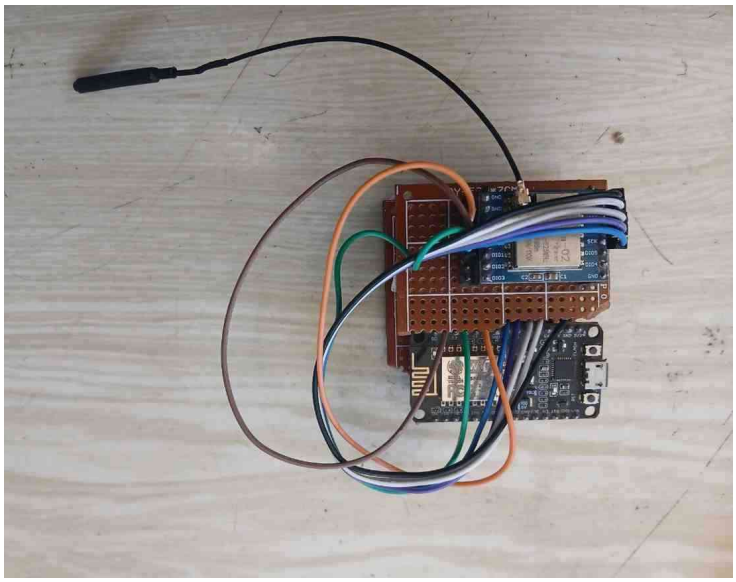

Figure 3.13: Prototype Receiver

3.5. Conversion of cartesian coordinates to Geographical Coordinates

The localized cartesian coordinates must be transformed into geographical coordinates in order to tell the UAS to fly to the nearby GPS location.Considering the latitude and longitude of the reference point, along with the distances to its east and north, the localized coordinates are computed using

$$\text{Latitude} = \text{Latitude}_{\text{ref}} + \frac{y \cdot 180}{6378000 \cdot \pi}$$

$$\text{Longitude} = \text{Longitude}_{\text{ref}} + \frac{x \cdot 180}{6378000 \cdot \pi \cdot \cos(\text{Latitude}_{\text{ref}})}$$

The formula, which is based on spherical trigonometry principles, is frequently used in programming languages and navigation.Values of x and y can be implemented immediately once the localized cartesian point (x,y) is obtained with respect to the reference point (0,0).

Since the cartesian coordinate system is fixed, directing the X-axis to the east and the Y-axis to the north, respectively, in the coordinate transformation formula.

3.6. Modality Sensing

The foundation of our localization model is the measurement of RSSI values as the drone hovers.Three modules make up the sensory modality.

- The first module is an SX1278 transceiver, which is attached to the antenna and has the ability to measure the RF source's RSSI.It uses the Serial Peripheral Interface (SPI) to connect to the microcontroller module.
- The second module is the Nodemcu ESP8266 microcontroller module.The RSSI values are obtained via the LORa module.The Raspberry Pi is the third module.Following the import and analysis of the RSSI values.
- It provides the Pixhawk with the instructions it needs to navigate by executing the localization calculation from the Nodemcu ESP8266.

The following illustrates the suggested modality sensing:

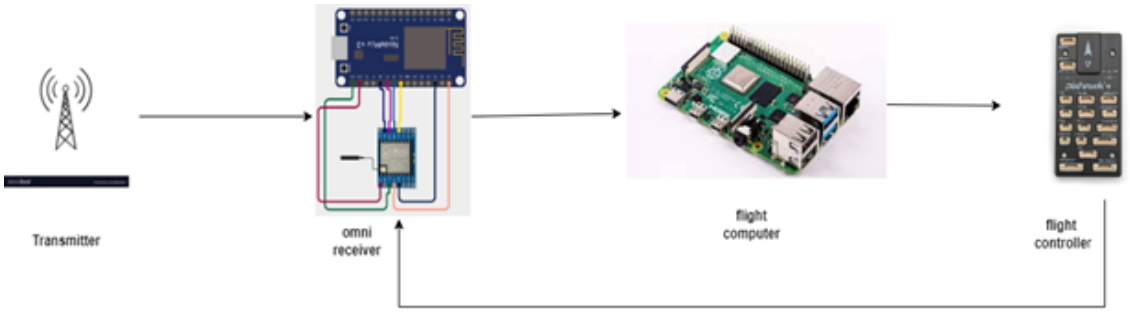

Figure 3.14: Modality Sensing

3.7. Model Validation

It is necessary to carry out an experiment with known parameters in order to verify the localization modal.The experiment will be carried out in a controlled,open setting.where the known position of the RF source will be installed.A drone operating the localization mode is launched, and its position must be compared to the precise location.The RF source position will be changed during the experiment a number of times, after which the modal's overall accuracy will be examined and confirmed.While checking, Geo Fence will also be set up to prevent unauthorized flying.

3.8. Antenna Calibration

3.8.1. Introduction

LoRa (Long Range) communication is a wireless technology widely used in low-power,long range applications such as environmental monitoring, smart cities, and RF source localization.The performance of LoRa communication is highly dependent on the antenna system,as the antenna directly influences the signal strength, coverage range, and reliability of received data.In RF based localization systems, especially those using Received Signal Strength Indicator (RSSI) measurements, antenna calibration plays a important role in ensuring the accuracy and consistency of the RSSI(received signal strength values).Proper calibration of the LoRa antenna is essential to minimize errors caused by hardware imperfections and environmental factors, ultimately enhancing the accuracy of the RSSI based localization process.

3.8.2. Characteristics

Antenna calibration is the process of determining and compensating for the non-ideal behavior of an antenna in order to obtain accurate signal strength measurements.Every antenna has unique characteristics such as gain, radiation pattern, and polarization, which can introduce systematic errors into RSSI measurements if not properly accounted for.Omnidirectional antennas at 433 MHz are a pragmatic choice for RSSI-based localization due to their simplicity, cost-effectiveness, and robustness in cluttered environments.
The primary antenna characteristics that require calibration are:

- Antenna Gain (G):The directional amplification of the signal by the antenna, typically expressed in dBi.Higher gain antennas amplify signals more in certain directions, which can introduce bias in RSSI readings depending on the relative orientation of the transmitter and receiver.
- Radiation Pattern:The three dimensional distribution of radiated power around the antenna.Omnidirectional antennas ideally provide uniform radiation in all directions, but practical antennas may exhibit variations that need to be corrected.
- Impedance Matching:Mismatch between antenna impedance and the transmission line can cause signal reflections, resulting in power loss and inaccurate RSSI readings.
- Polarization Mismatch:The alignment of the electric field between the transmitter and receiver antennas can affect the received power.Misalignment can lead to attenuation of the signal.

3.8.3. Importance of Antenna Calibration in RSSI-Based Localization

RSSI-based localization techniques rely heavily on the accuracy and consistency of the measured signal strength.Uncalibrated antennas can introduce systematic errors in the measured RSSI values, leading to significant inaccuracies in distance estimation and position calculations.
The key benefits of antenna calibration in RSSI-based localization are:

1. Improved Distance Estimation:Accurate RSSI values result in better calculating the distance between the transmitter and receiver using the path loss model formula.
2. Minimized Systematic Errors:Calibration reduces hardware induced biases that would otherwise skew the localization results.

3. Consistency Across Measurements:Proper calibration ensures that RSSI readings remain consistent across different antennas and measurement sessions.

4. Robustness in Environmental Variations:Environmental factors such as temperature and humidity can affect antenna performance, and calibration helps mitigate these variations.

5. Enhanced Multilateration Accuracy:In multilateration algorithms, small deviations in RSSI data can lead to large localization errors.Calibrated antennas significantly improve the precision of the final estimated coordinates.

3.8.4. Mitigation Strategies

1. To enhance localization accuracy with 433 MHz omnidirectional

2. Multi Antenna Arrays:Use multiple omnidirectional antennas in a phased array to estimate directionality.

3.9. Path-loss Exponent(PLE)

Path Loss Exponent (PLE) is a critical parameter in wireless communication systems, particularly in the context of signal propagation modeling.It quantifies the rate at which the signal power decays with distance in a given environment.It is denoted by (dBm).

Path loss refers to the reduction in power density of an electromagnetic wave as it propagates through space.It is a fundamental concept in wireless communication and is influenced by factors such as distance, frequency, and environmental conditions (e.g.urban,suburban, rural). Path Loss Exponent (n) The path loss exponent (n) characterizes the rate at which the signal power decreases with distance.Its value depends on the propagation environment.

- Free Space: $n = 2$.
- Urban Areas: $n = 2.7$ to 3.5.
- Suburban Areas: $n = 3$ to 4.
- Indoor Environments: $n = 4$ to 6.

3.9.1. Factors affecting Path Loss exponent:

1. **Environment**:
    - Open areas (e.g.,free space) have lower n.
    - Dense urban or indoor environments have higher n.
2. **Frequency**:
    - Higher frequencies experience greater path loss.
3. **Antenna Characteristics**:
    - Antenna height, gain, and orientation affect signal propagation.
4. **Obstacles**:
    - Buildings, trees, and walls increase n.

3.9.2. Log-distance path loss model

The most common path loss model is the log-distance path loss model, which is given by:

$$PL(d) = PL(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma$$

Where:

- $PL(d)$: Path loss at distance d (in dB).
- $PL(d_0)$: Path loss at a reference distance d_0 (in dB).
- n: Path loss exponent (dimensionless).
- d: Distance between transmitter and receiver (in meters).
- d_0: Reference distance (typically 1 m or 1 km).
- X_σ: A zero-mean Gaussian random variable representing shadowing effects (in dB).

3.10. Omnidirectional Antenna

Omnidirectional antenna is a type of antenna that radiates or receives electromagnetic waves uniformly in all directions in a single plane.For LoRa (Long Range) communication modules operating at 433 MHz, omnidirectional antennas are widely used due to their ability to provide consistent coverage in all directions.

3.10.1. Characteristics of Omnidirectional Antenna

1. **Radiation Pattern**:
    - Omnidirectional antennas have a doughnut-shaped radiation pattern in 3D space.
    - Maximum radiation occurs in the horizontal plane.
    - Minimal radiation occurs in the vertical direction.
    - Suitable for applications where devices are spread out horizontally (e.g., sensors in a field).
2. **Gain**:
    - Omnidirectional antennas typically have low to moderate gain (e.g., 2 to 5 dBi).
    - The gain is achieved by compressing the vertical radiation pattern.
    - This increases the horizontal coverage.
3. **Polarization**:
    - Most omnidirectional antennas for 433 MHz are linearly polarized.
    - Vertical polarization is common.

3.11. Filtering

Filtering is the process of refining datasets by including or excluding data points based on specific criteria.It ensures that only relevant, accurate,and high quality data is retained for analysis.Techniques range from removing outliers and duplicates to enforcing format consistency or logical constraints.

3.11.1. Introduction to Filtering

Filtering is a signal processing technique used to extract useful information from noisy or corrupted signals by removing unwanted components such as noise or interference.In wireless communication systems, signal measurements are often subject to environmental noise, hardware imperfections, and unpredictable variations.Filtering plays a vital role in improving the accuracy and reliability of the measured data, especially in applications where precision is critical, such as RF source localization using Received Signal Strength Indicator (RSSI).

3.11.2. Noise in RSSI Measurements

RSSI is a commonly used parameter for estimating the distance between a transmitter and a receiver in RF localization systems.However, RSSI measurements are highly susceptible to noise due to several factors, including:

1. Multipath Propagation:Reflections of the RF signal from obstacles like buildings, walls, or trees can cause signal fluctuations.
2. Environmental Factors:Temperature, humidity, and interference from other RF devices affect the received signal strength.
3. Hardware Imperfections:Variations in the sensitivity of RF modules and antenna gains introduce noise into the measurements.

These factors make the raw RSSI data highly inconsistent and unreliable, which can significantly degrade the accuracy of RF source localization if not properly filtered.

3.11.3. Importance of Filtering in RSSI-Based Localization

1. Filtering techniques are applied to mitigate the noise and fluctuations in RSSI measurements,enhancing the stability and precision of the estimated RF source location.
2. The primary objectives of filtering in RSSI based localization are.
3. Noise Reduction:Suppress random fluctuations and high-frequency noise in RSSI readings.
4. Data Smoothing:Provide a more stable and continuous RSSI signal over time.

5. Outlier Removal:Eliminate sudden, unexpected signal variations caused by interference or measurement errors.

6. Improved Estimation Accuracy:Enhance the performance of localization algorithms such as multilateration by providing cleaner input data.

3.11.4. Simple Moving Average Filter

The simple moving average is a filtering technique that generates a new series of data points by averaging a specified group of data referred to as a 'window' and shifting this window across a time series.Each data point within the window carries the same importance, implying that they all equally influence the average.While extending the length of the window yields a more smoothed result, it also causes the filtered data to lag further.This filter is ideal for real-time applications due to its ease of implementation and minimal resource consumption.An SMA filter with a window size of 'n' can be employed as every data point in RSSI filtering receives equal importance.The SMA filter necessitates a number of data points to compute the average.Initially, there are not enough data points in the time series to achieve this, resulting in the output being either assigned a default value or remaining unspecified.Once there are adequate data points to fill the window, the filter can calculate the average of the data subset and produce an output sequence.It emphasizes longer-term trends in the data while smoothing out short-term fluctuations.The operation involves averaging a predetermined number of past data points in a time series.

For a given time series $x\{n\}$, the SMA at time n with a window size of N is:

$$SMA\{n\} = \frac{1}{N} \sum_{i=0}^{N-1} x\{n-i\}$$

Where:

- $x\{n\}$ is the input signal or data series,
- N is the window size (number of past points considered),
- $SMA\{n\}$ is the smoothed output at time n.

3.12. Software in the Loop (SITL)

Software-in-the-Loop (SITL) simulation allows the PX4 flight code to operate a computer modeled vehicle within a virtual environment.By running the full PX4 software stack on a

computer,SITL creates a simulated environment that mimics the behavior and responses of a real vehicle.We use simulators Gazebo to generate realistic environmental data and sensor inputs.In this virtual environment, a ground control station typically QGroundControl communicates with the simulated vehicle via the MAVLink protocol.This enables users to interact with and command the simulated drone just as they would with an actual vehicle.
In SITL,specialized firmware runs in a virtual context,processing simulated sensor data (including inertial measurements, GPS signals, and optical flow) while generating corresponding actuator outputs.This configuration not only offers a safe and efficient platform for testing and debugging flight algorithms but also permits rapid iteration and integration before deployment on hardware.Consequently,SITL has become a critical tool for validating main mission flight code and ensuring system reliability during early development stages.

Finally, we have completed the SITL simulation of our main mission flight code.

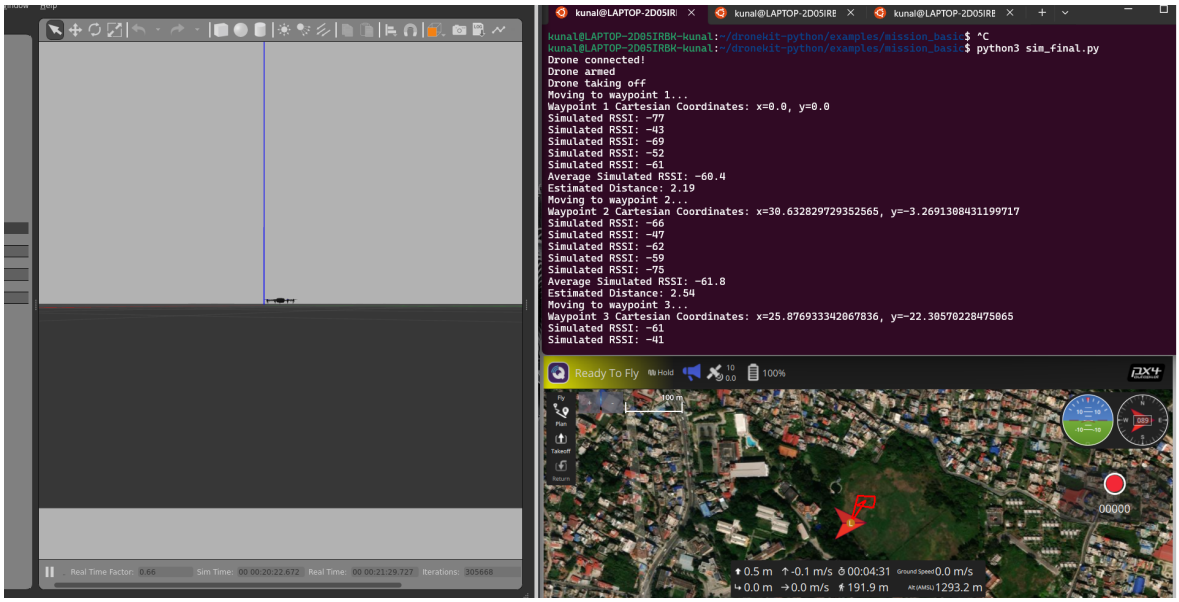

Figure 3.15: Software-In-The-Loop (SITL) Simulation Setup

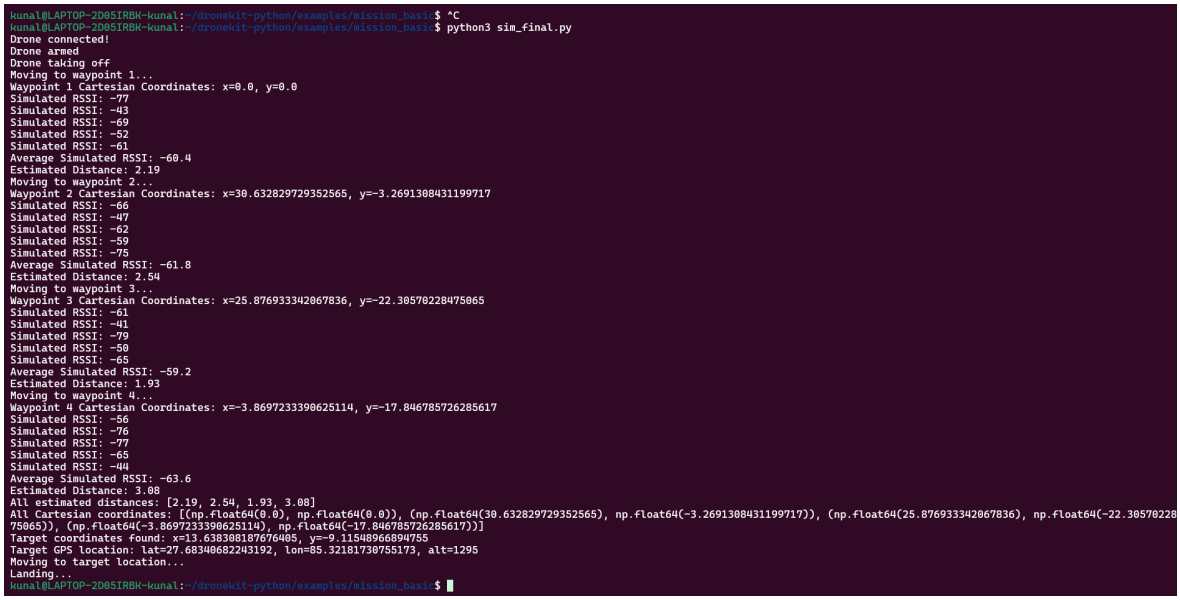

Figure 3.16: Software-In-The-Loop (SITL) Output terminal

3.13. Hardware in the Loop (HITL)

In our experimental setup, Hardware In The Loop (HITL) simulation is employed to rigorously validate the mission code using an actual Pixhawk 4 flight controller.The Pixhawk 4, running PX4 firmware, is physically integrated into the system via a USB cable connection to the Gazebo Classic simulator.This connection enables Gazebo Classic to serve as an intermediary conduit, facilitating the bidirectional exchange of MAVLink messages between PX4 and QGroundControl.As a result,the simulator accurately emulates real world flight dynamics and communication, allowing for comprehensive testing and refinement of the mission code under conditions that closely mimic actual flight operations

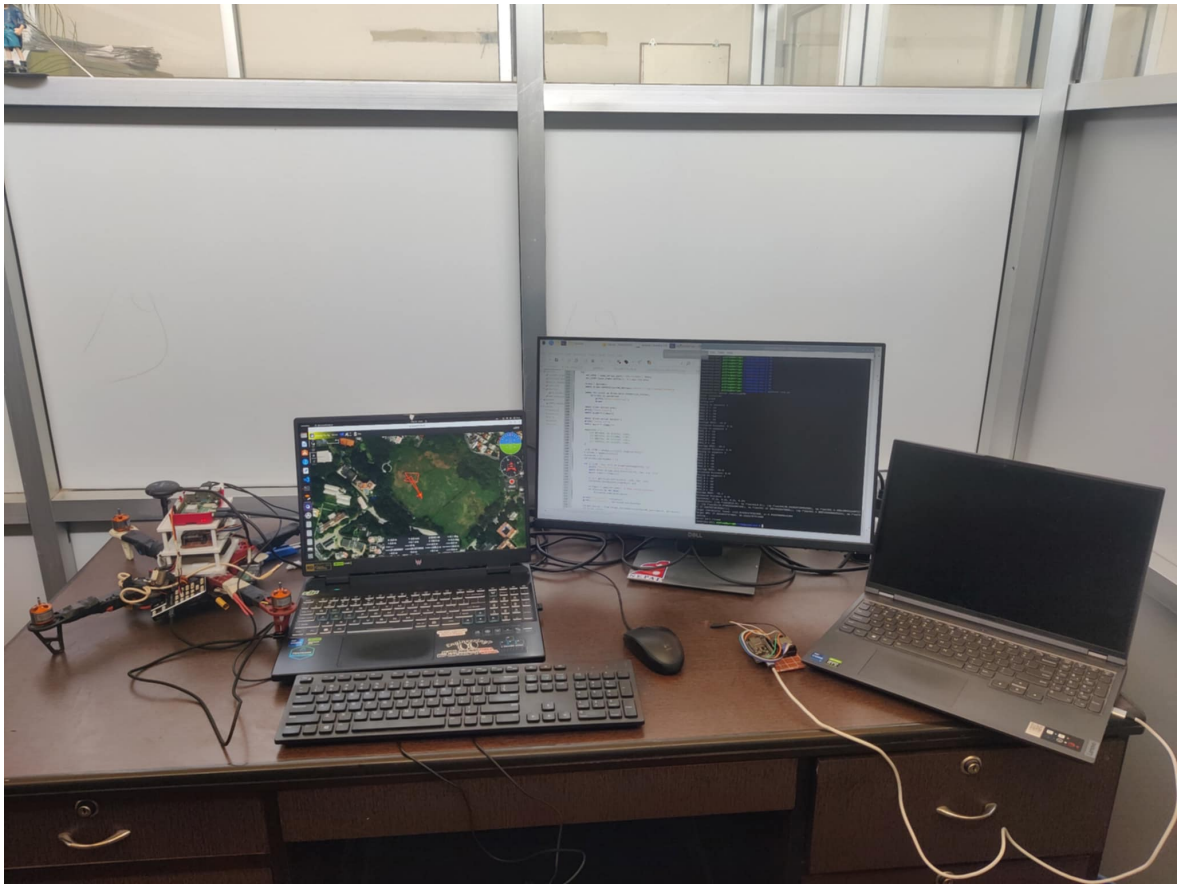

Figure 3.17: Hardware-In-The-Loop (HITL) Simulation Setup

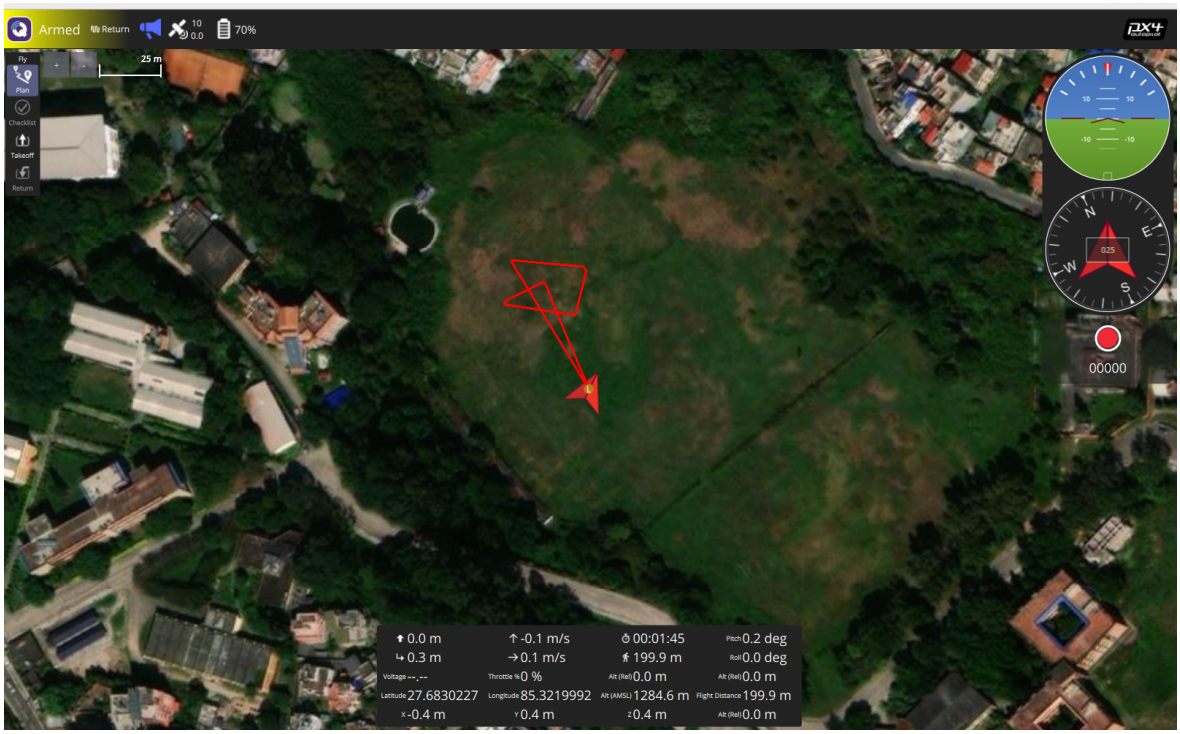

Figure 3.18: Hardware-In-The-Loop (HITL) Simulation Output

3.14. Autonomous Drone

In this project, we develop a method for localizing an RF source using an autonomous drone.The drone's autonomy is facilitated by employing a Raspberry Pi 4 as its companion (flight) computer.The Raspberry Pi 4 gathers sensor information, processes it, and produces the necessary control commands for the drone's autonomous navigation.These commands are transmitted to the Pixhawk flight controller, which then modifies the drone's position, orientation, and movement accordingly.The main libraries utilized in our primary mission code include MAVSDK and pymavlink:
The key libraries used in our main mission code are MAVSDK and pymavlink:

1. MAVSDK:A modern, high-level API (Application Programming Interface) that provides an intuitive interface to communicate with the Pixhawk flight controller.It simplifies the development of autonomous drone applications by offering user-friendly functions for mission planning,telemetry data retrieval, and command execution, enabling seamless integration of autonomous features.

2. pymavlink:A Python implementation of the MAVLink protocol, which facilitates low level communication between the Raspberry Pi and Pixhawk.It ensures reliable transmission of commands and telemetry data, allowing real-time flight parameter monitoring and custom control commands.

This combination of libraries enables the autonomous drone to perform waypoint navigation, collect sensor data, and execute localization tasks efficiently.

3.15. Autonomy level of UAV

The autonomy level of an UAV is basically defined as the ability to operate without the need of human input.Autonomy levels range from manual control to full autonomy, where the UAV can complete its mission without human input.These levels can be categorized as follows:

Level 0: No Autonomy (Manual Control)

The UAV is fully controlled by a human pilot using a remote control or joystick.
Example: Traditional RC drones.

Level 1: Pilot Assistance (Basic Automation)

The UAV has stabilization and basic assistance features like altitude hold.
Example: Drones with GPS hold or basic autopilot functions.

Level 2: Partial Autonomy (Waypoints Navigation)

The UAV can follow predefined waypoints but still requires human supervision.
Example: Mission planning with GPS-based waypoint following in ArduPilot/PX4.

Level 3: Conditional Autonomy (Decision Support)

The UAV can make limited decisions like obstacle avoidance and path adjustments.
Still requires human intervention in complex situations.
Example: Drones with real-time obstacle detection (e.g., DJI AirSense).

Level 4: High Autonomy (Minimal Human Input)

The UAV can perform complex missions independently, including real-time adaptation to dynamic environments.
Can communicate with other UAVs or systems but might need occasional human oversight.
Example: Swarm UAVs performing cooperative tasks, delivery drones with dynamic rerouting.

Level 5: Full Autonomy (No Human Involvement)

The UAV can plan, execute, and complete missions without human intervention.
Can make high-level decisions and adapt to unknown situations.
Example: Fully autonomous surveillance UAVs, AI-driven drones for emergency response.

From the analysis of different levels of autonomy of the UAV,our UAV operates fully autonomously from takeoff to landing once we run the Python script. Since there are no manual inputs during mission flight,and it follows a predefined set of actions (waypoints navigation),and records RSSI data,find localized coordinate, apply multilateration formula, source hovering, and finally back to return.Therefore, the autonomy level of our UAV is level 3 (Conditional Autonomy).

3.16. Flight Data Analysis (Log File)

The log file in Pixhawk refers to the flight data recorded by the Pixhawk flight controller during a flight.These logs are essential for analyzing the performance of UAS, diagnosing issues,and understanding the behavior of the autopilot system. Pixhawk logs contain detailed information about the vehicle's state, sensor data, control inputs, and more.

3.16.1. Sensor Data

- Accelerometer, gyroscope, and magnetometer readings.
- GPS position, velocity, and satellite information.
- Barometer (altitude) data.

3.16.2. Vehicle State

- Attitude (roll, pitch, yaw).
- Position (latitude, longitude, altitude).

3.16.3. Control Inputs

- RC (remote control) inputs.
- Actuator outputs (motor and servo commands).

3.16.4. Autopilot Behavior

- Flight mode changes.
- Waypoint navigation data.
- Error and status messages.

3.16.5. System Information

- Battery voltage and current.
- CPU load and memory usage.
- Error codes and warnings.

CHAPTER 4: RESULTS AND DISCUSSION

4.1. Indoor Localization

The localization in indoor environment of RF (Radio Frequency) sources involves the process of determining the accurate position of a RF source within an indoor environment using RF signals.

4.1.1. Challenges in Indoor Localization

1. Multipath propagation,
2. Signal attenuation
3. Interference from walls
4. Obstacles.

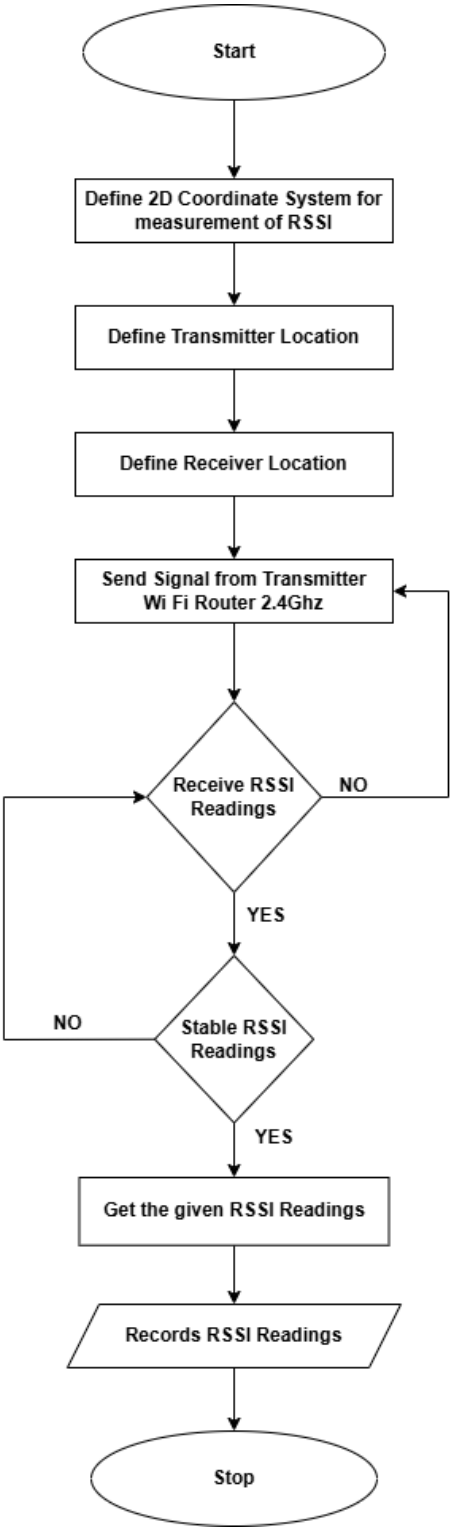

Figure 4.1: Indoor test flowchart

4.1.2. Experimental Result

The indoor localization was done in Seminar hall of the Center of Energy Studies at Pulchowk Campus.The radio source was a 2.4 GHz WiFi router.Four predetermined positions were used for measurements after the setup environment had been calibrated, and the mean RSSI values for each position were calculated.The coordinates system utilized for the indoor localization is displayed in figure 4.3.

Figure 4.2: Indoor Localization Space

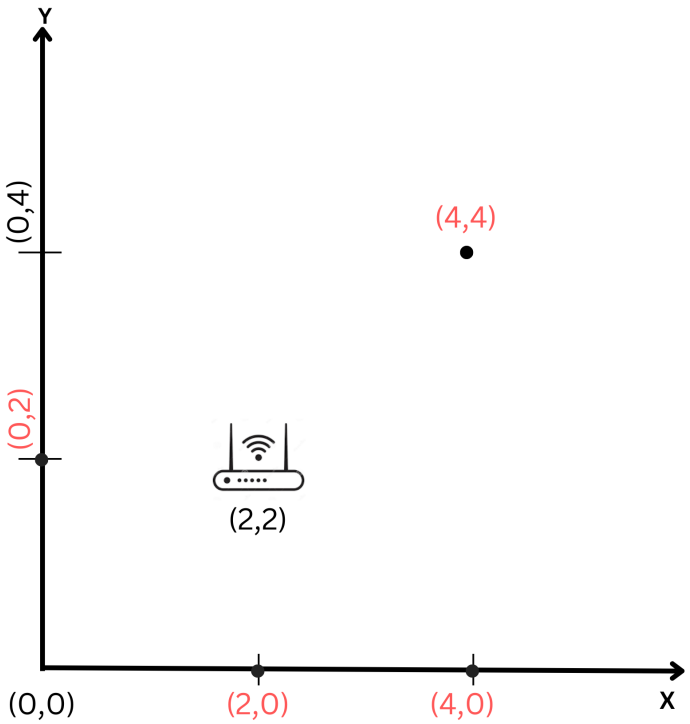

Figure 4.3: Indoor Coordinate System

For calibration of the RSSI vales, signal strength values were measured at four different locations from the transmitter.The corresponding mean RSSI values at these locations are tabulated in table below:

Table 4.1: Experimental Data of Indoor Localization

Coordinates	Distance(m) from Transmitter	Mean RSSI (dBm)	Standard deviation
(2,0)	2	-53.93	5.70
(0,2)	2	-47.84	2.33
(4,0)	2.82	-50	4.71
(4,4)	2.82	-51.67	1.93

from the above data points, the log-distance path loss model is fitted.The value for the path loss exponent, n, is obtained to be 4.7275 and the localized coordinate is (1.9549,2.0598).

Table 4.2: The Localized Data of Indoor Localization

Test	Localized Cartesian		Distance error (m)
	X	Y	
1	1.9549	2.0598	0.0749

4.1.3. Matlab Simulation

1. Path Loss Model Techniques:An omnidirectional transmitter located at (250,250) is modeled in 2-dimensional space of 500 m by 500 m.The Received Signal Strength Indicator (RSSI) at a distance d is given by the following equation:

$$RSSI = RSSI_0 - 10n\log_{10}\left(\frac{d}{d_0}\right) - X(\sigma)$$

Where:

- $RSSI$ is the received signal strength at distance d.
- $RSSI_0$ is the received signal strength at a reference distance d_0.
- n is the path loss exponent, which depends on the environment (e.g., free space, urban, indoor).
- d is the distance between the transmitter and receiver.
- d_0 is a reference distance where $RSSI_0$ is measured.
- $X(\sigma)$ represents the shadowing or fading effect, modeled as a random variable with a Gaussian distribution, mean 0, and standard deviation σ.

A noise model known as Additive White Gaussian Noise (AWGN), based on Gaussian distributed random variable is used to mimic the random noises that occur in nature.

4.1.4. The simulation model's capabilities:

- Using the path loss model and related parameters, it calculates the RSSI values at each grid point.
- Enables the user to measure matching RSSI values by entering three distinct places.
- Converts those RSSI readings into the appropriate distance measurements.
- Determines the transmitting device's coordinates by solving a system of three non-linear equations.

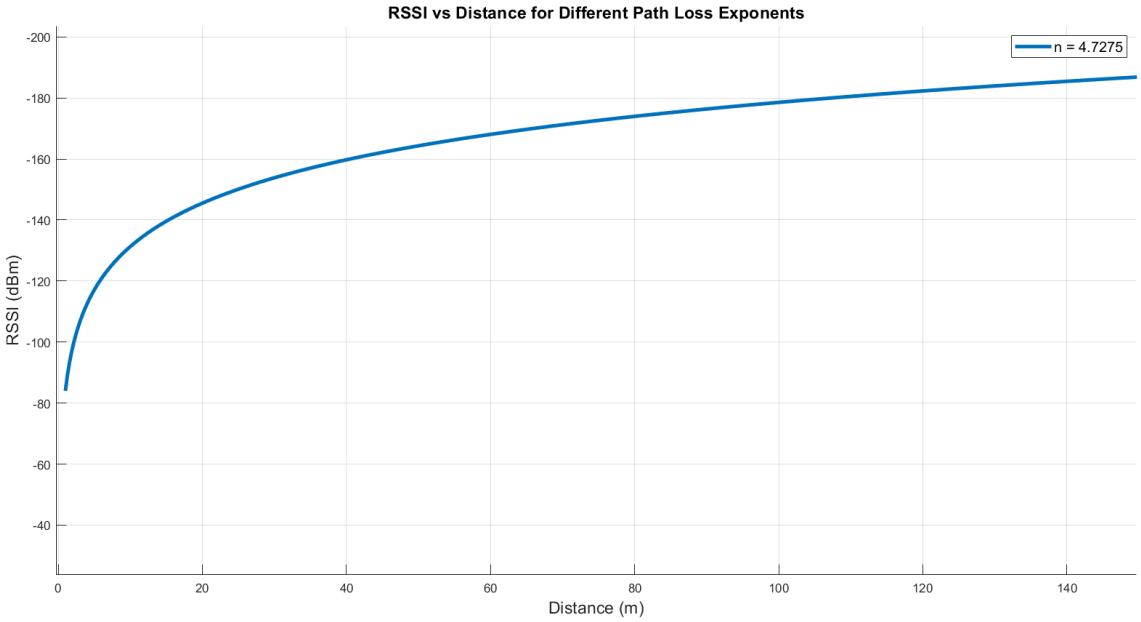

Figure 4.4: RSSI VS DISTANCE (Without AWGN modeling)

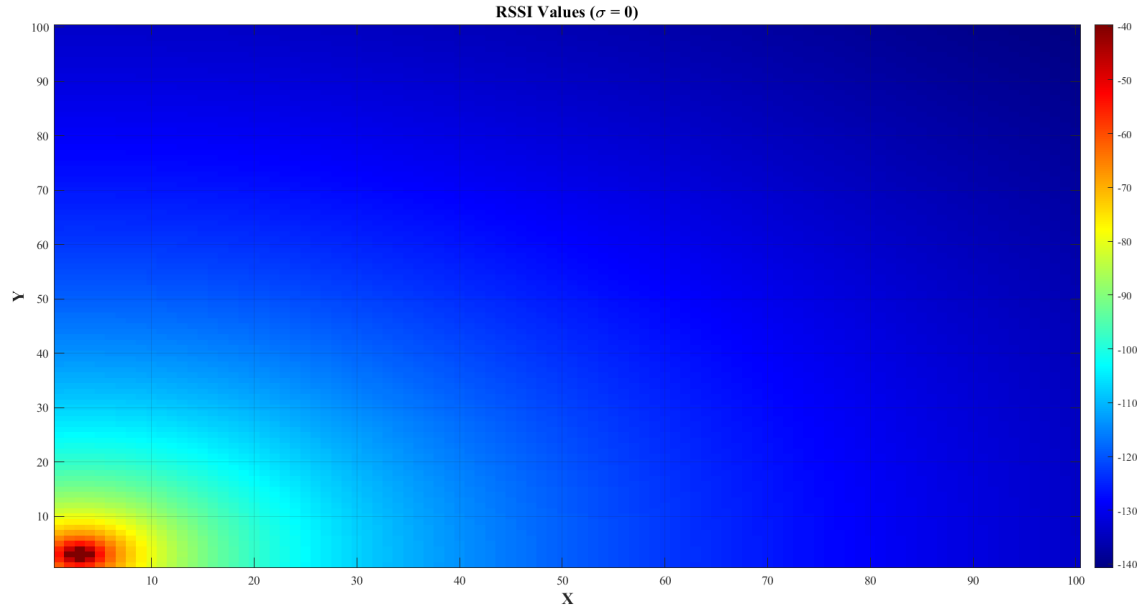

Figure 4.5: RSSI Value ($\sigma = 0$)

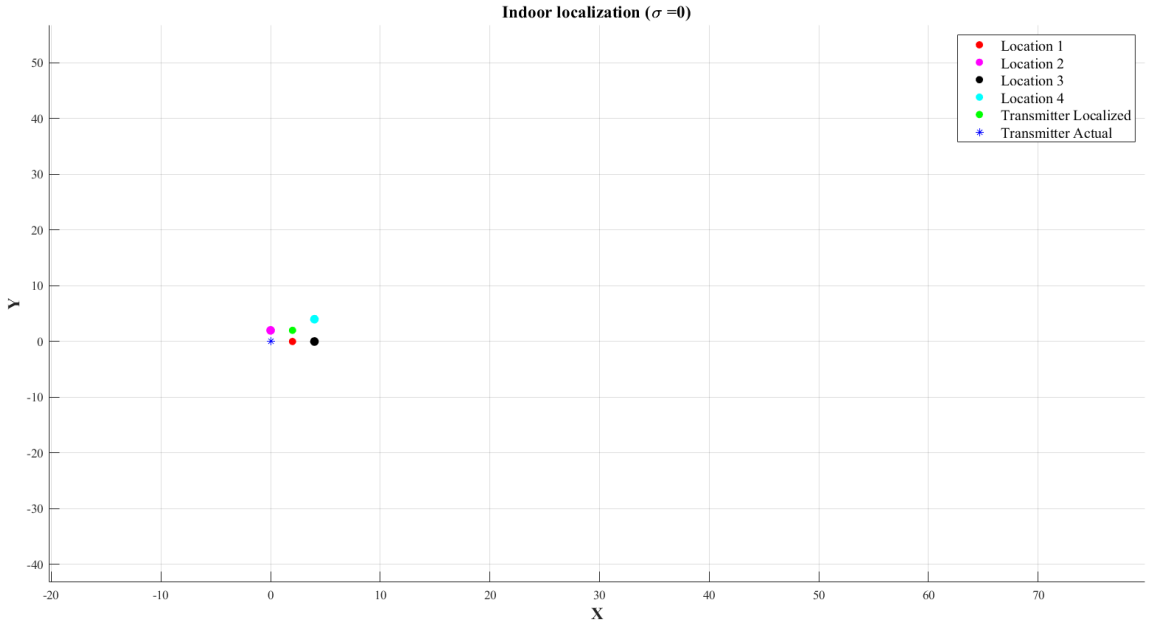

Figure 4.6: Indoor Localization ($\sigma = 0$)

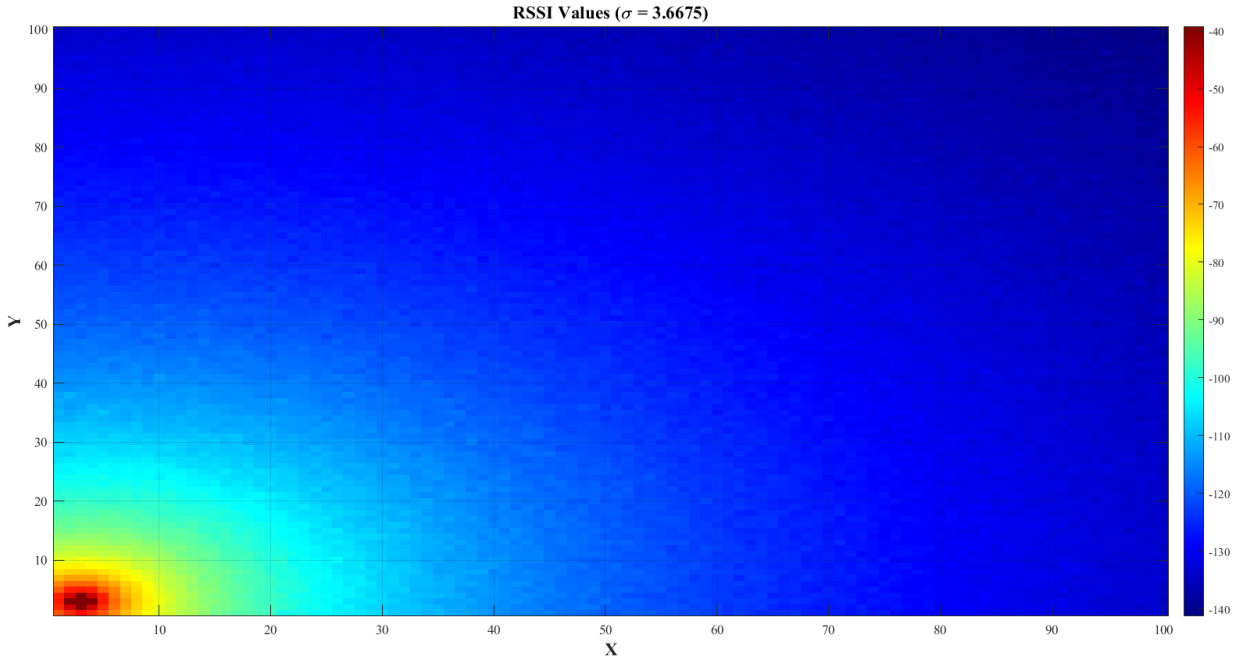

Figure 4.7: RSSI Value ($\sigma = 3.6675$)

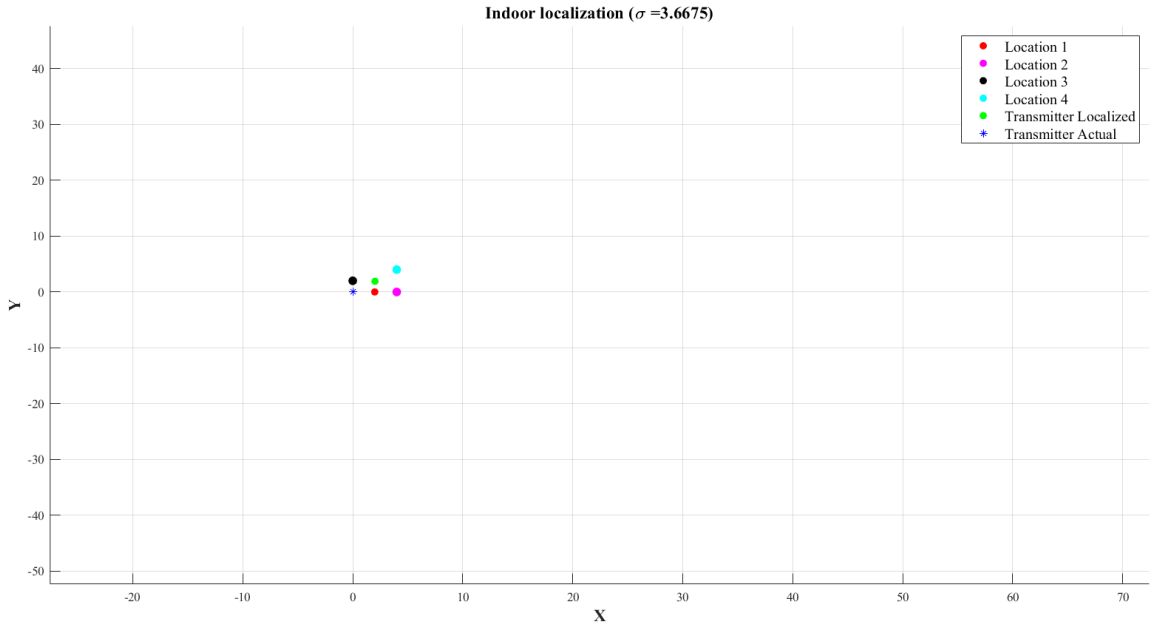

Figure 4.8: Indoor Localization ($\sigma = 3.6675$)

4.2. Outdoor Localization

Outdoor Localization denotes the method of identifying the exact geographic position of an object, device, or signal source within an outdoor atmosphere.

4.2.1. Challenges in Outdoor Localization

- Environmental Factors:Weather, terrain, and obstacles (e.g., buildings, trees) can interfere with signals and reduce accuracy.
- Signal Interference:Competing signals or noise can distort measurements.
- Energy Consumption:Continuous localization can drain battery-powered devices.
- Scalability:Extending localization to large areas or multiple targets increases complexity.

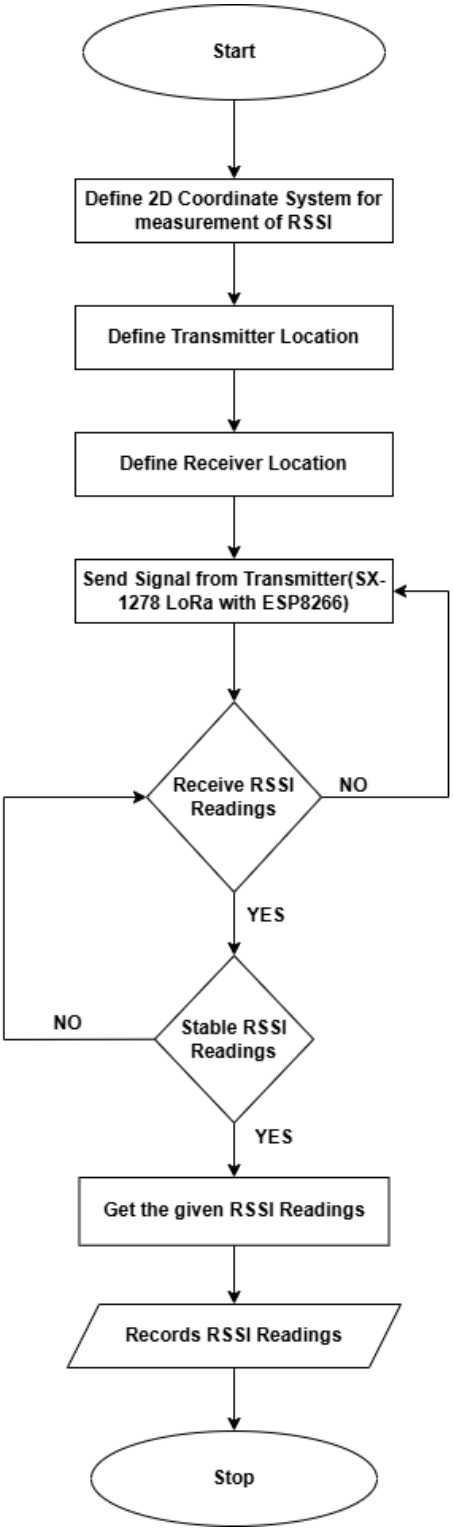

Figure 4.9: Outdoor test flowchart

4.2.2. Experimental Result

The transmitter was positioned in a exact spot on the campus cricket ground, and the receivers were dispersed throughout several known points.The datawas collected from the LoRa module.The unique path loss model was calibrated using the processed RSSI values.Following a proper and precise calibration, the multilateration process was used to locate the radio source after measuring RSSI values at four known locations.Calibration of outdoor environment was carried and the data is tabulated below:

Table 4.3: Calibration data

Distance(m) from Transmitter	Mean RSSI (dBm)	Standard deviation
1	-74.39	2.72
5	-87.46	1.41
10	-94.84	2.28
20	-98.73	3.48
30	98.93	2.46
40	-99.69	2.58

from the above data,the value of path loss exponent,n, comes to be 2.062.

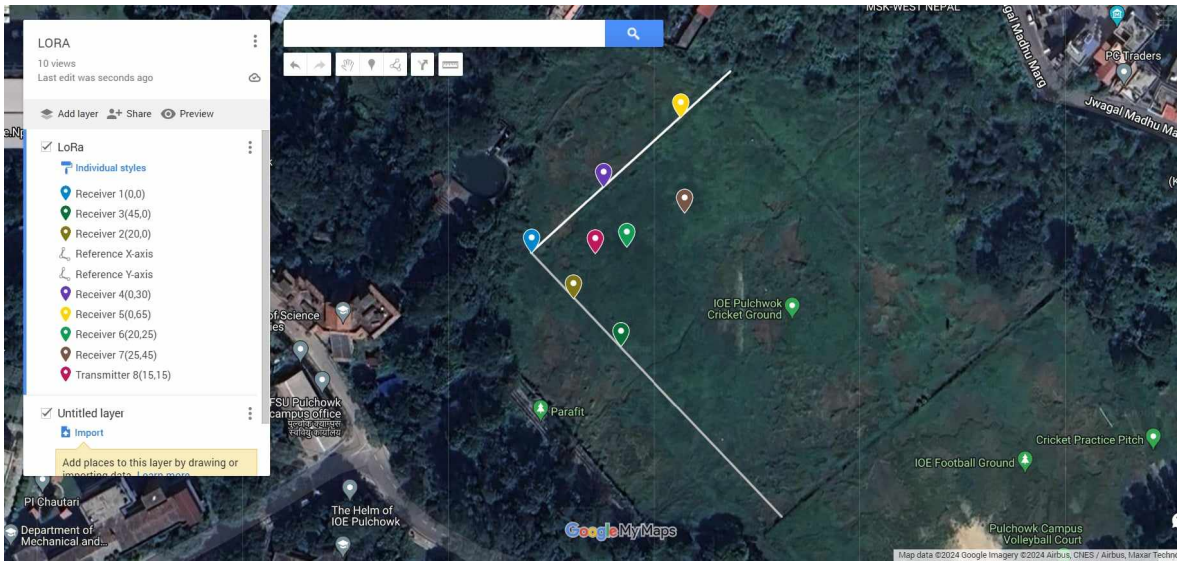

Figure 4.10: LoRa test result at Campus

We performed the Outdoor Localization test without using drone setup keeping transmitter at (15,15) and the data obtained from the experiment is shown below:

From table 4.4, the path loss exponent comes to be 2.85 and taking four coordinates (0,0),(0,30), (45,0),(25,45) ,for code verification we get the localized coordinates to be (14.8244,14.6135).

Table 4.4: Experimental data of outdoor Localization

Coordinates	Distance(m) from Transmitter	Mean RSSI (dBm)	Standard deviation
(20,25)	11.18	-88.98	2.145
(20,0)	15.81	-96.41	2.62
(0,0)	21.21	-94.87	3.10
(0,30)	21.21	-100.64	1.66
(25,45)	31.62	-99.98	1.94
(45,0)	35.54	-98.25	2.98
(0,65)	52.50	-98.45	3.10

Table 4.5: Outdoor Localization D

Test	Localized Cartesian		Distance error (m)
	X	Y	
1	14.8244	14.6135	0.42

4.2.3. Matlab Simulation

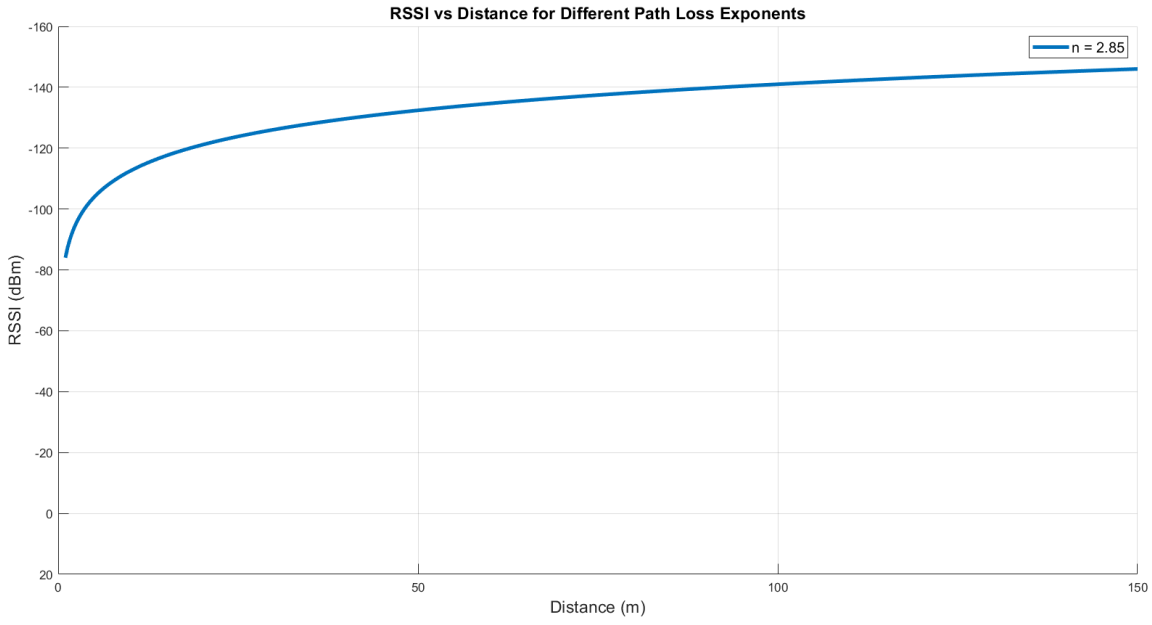

Figure 4.11: RSSI Value VS distance(Without AWGN modeling)

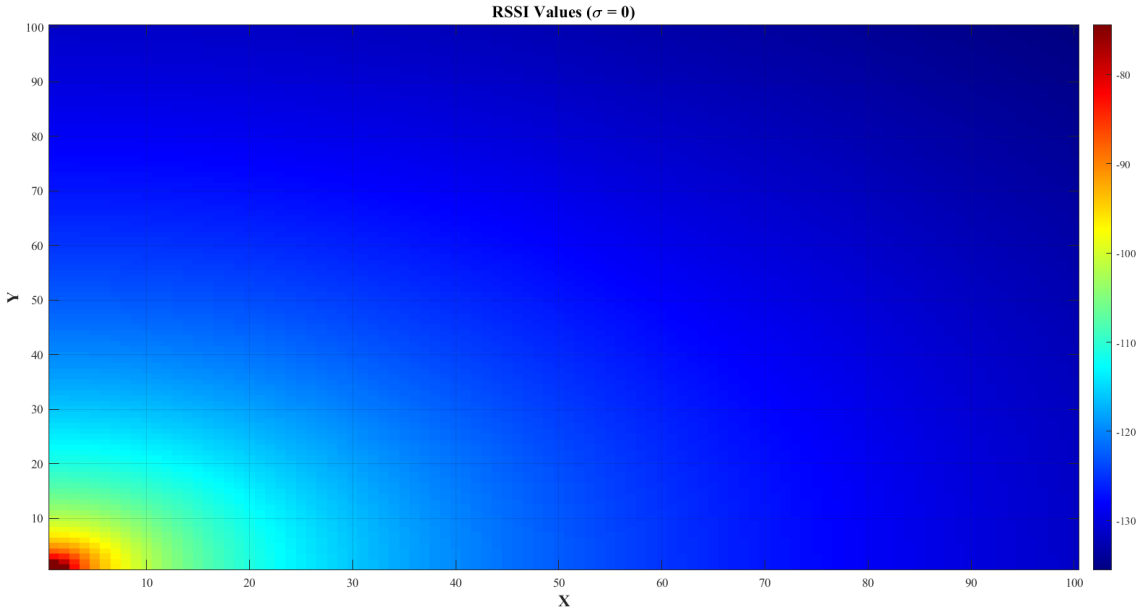

Figure 4.12: RSSI Value($\sigma = 0$)

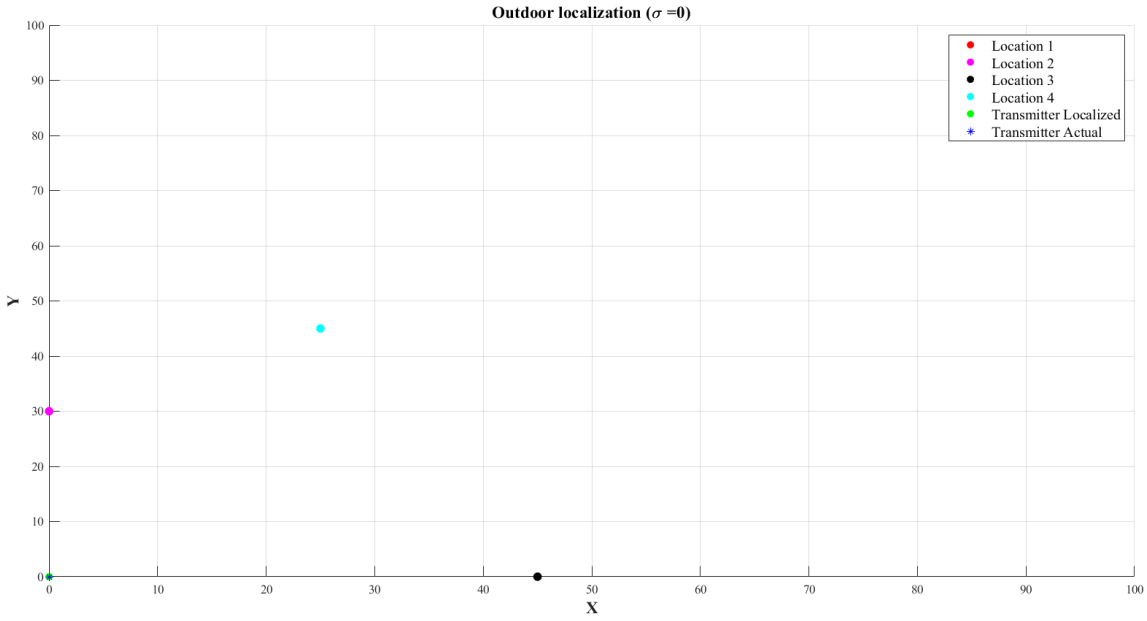

Figure 4.13: Outdoor Localization($\sigma = 0$)

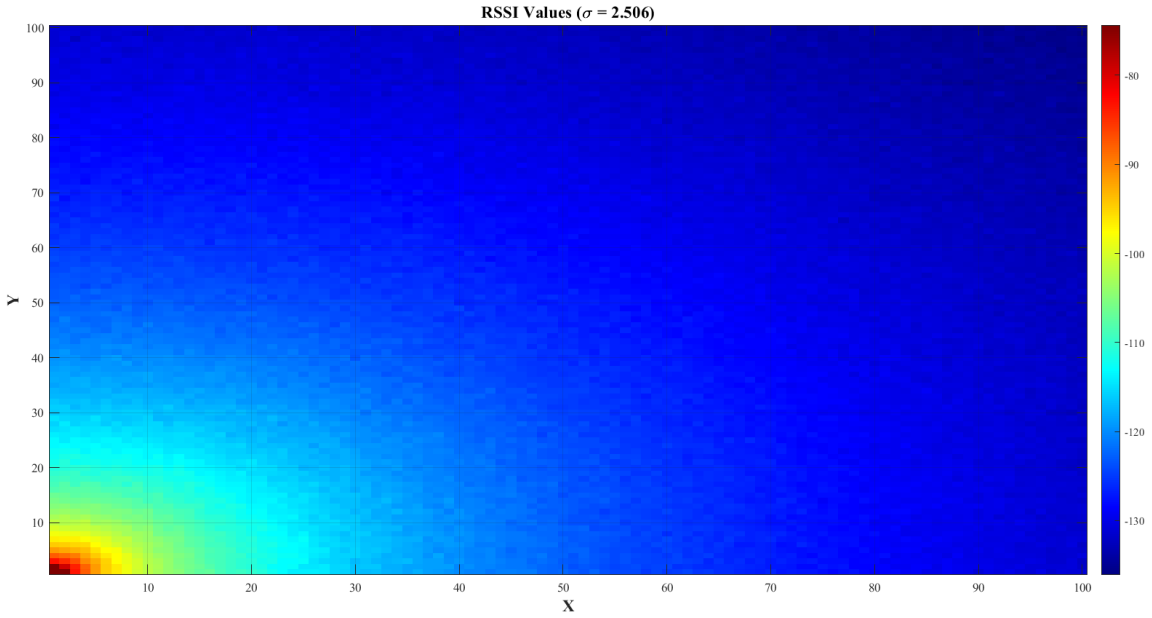

Figure 4.14: RSSI Value($\sigma = 2.506$)

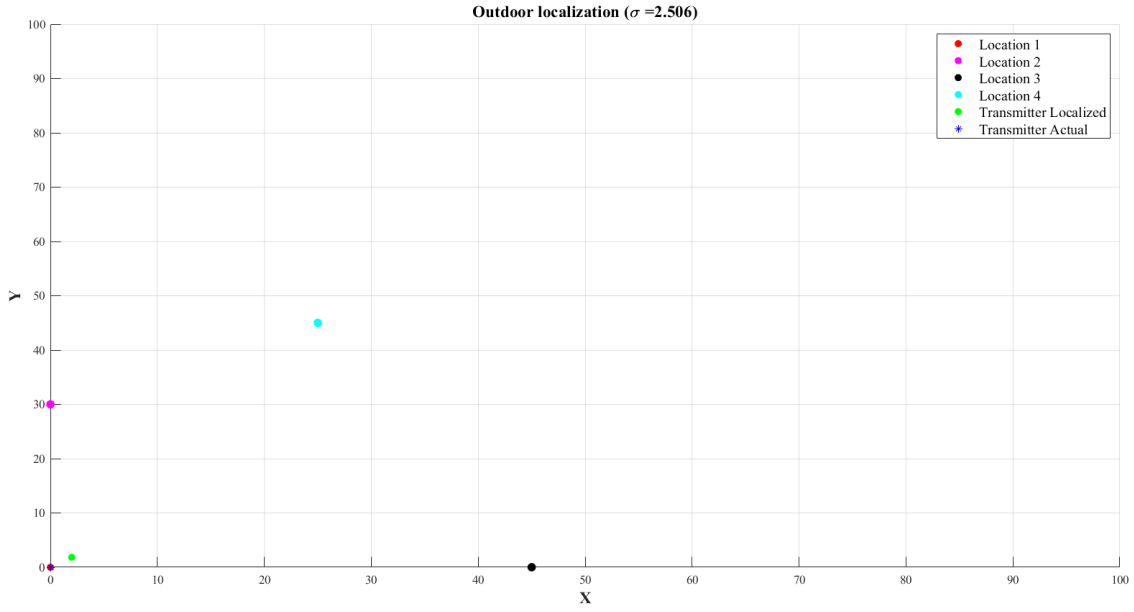

Figure 4.15: Outdoor localization($\sigma = 2.506$)

4.2.4. Antenna Calibration

Table 4.6: Antenna Calibration

Distance(m)	RSSI(dBm)
90	-118
80	-115
70	-110
60	-115
50	-117
30	-108
20	-103
10	-96
1	-77

From the above calibration data,the pathloss exponent are given by the log normal model,which is one of the most widely used distance estimation models for radio wave propagation, is given by

$$RSSI = RSSI_0 - 10n\log_{10}\left(\frac{d}{d_0}\right) - X(\sigma) \tag{4.1}$$

The effect of $X(\sigma)$ can be mitigated to some extent through the process of filtering.If we neglect $X(\sigma)$, and take the reference distance d_0 equal to 1 meter, then equation 4.1 becomes:

$$RSSI = RSSI_0 - 10n\log_{10}(d) \tag{4.2}$$

the path loss exponent(n),comes to be 2.053 from the above log normal model formula.

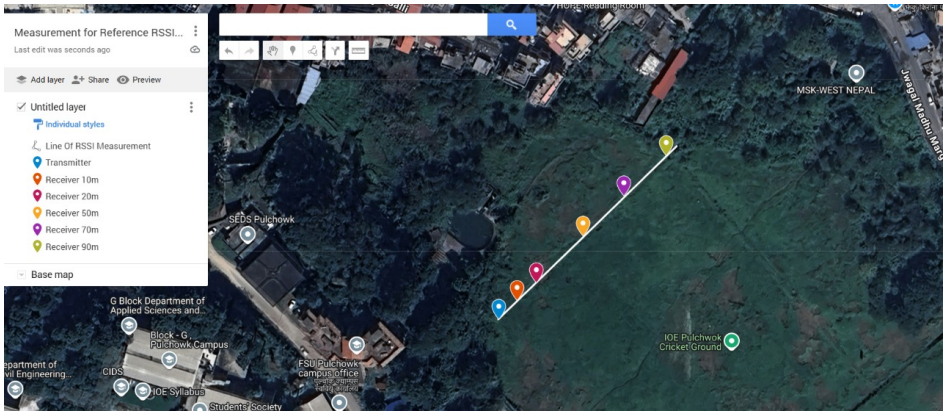

Figure 4.16: Antenna calibration at Pulchowk Campus

From the antenna calibration,we calculated the following distance errors:

Table 4.7: Antenna Calibration

Actual Distance(m) from Source	Estimated Distances	Distance Error
90	99.32	9.32
80	79.3704	0.6296
70	64	6
60	56.693	3.307
50	45.3015	4.6985
30	32.35	2.35
20	18.4688	1.5312
10	8.42	1.58

4.3. LoRa Test at TU,Kirtipur

We performed the Outdoor Localization test without using drone setup keeping transmitter at (0,0) and the data obtained from the experiment is shown below:

From table 4.7,the path loss exponent comes to be 2.1298 and taking four coordinates(0,0),(0,60), (0,40),(75,0),(47.5,64.5), for code verification we get the localized coordinates to be (0.000000009271,0.000000075346).

Table 4.8: Experimental Data of Outdoor Localization at TU,Kirtipur

Coordinates	Distance(m) from Transmitter	Mean RSSI (dBm)	Path loss Exponent
(0,0)	1	-84	
(75,0)	75	-123	2.09
(0,40)	40	-117	2.059
(0,60)	60	-122	2.139
(47.5,64.5)	80.10	-123	2.048
(35,25)	43.01	-122	2.362

Table 4.9: The Localized Data of Outdoor Localization($\sigma = 0$),TU Kirtipur

Test	Localized Cartesian		Distance error (m)
	X	Y	
1	0.000000009271	0.000000075346	1.19466e-10

Table 4.10: The Localized Data of Outdoor Localization($\sigma = 1.789$),TU Kirtipur

Test	Localized Cartesian		Distance error (m)
	X	Y	
1	0.0732123	0.56768	0.5724

Figure 4.17: LoRa test result at TU Kirtipur

4.3.1. Matlab Simulation

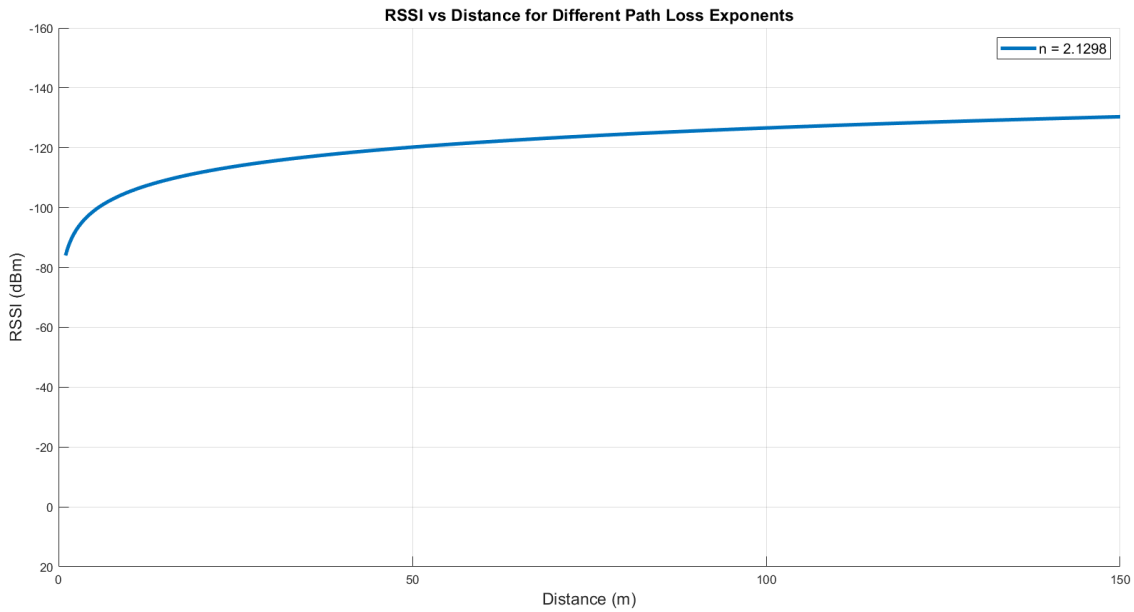

Figure 4.18: RSSI Value VS distance(Without AWGN modeling)

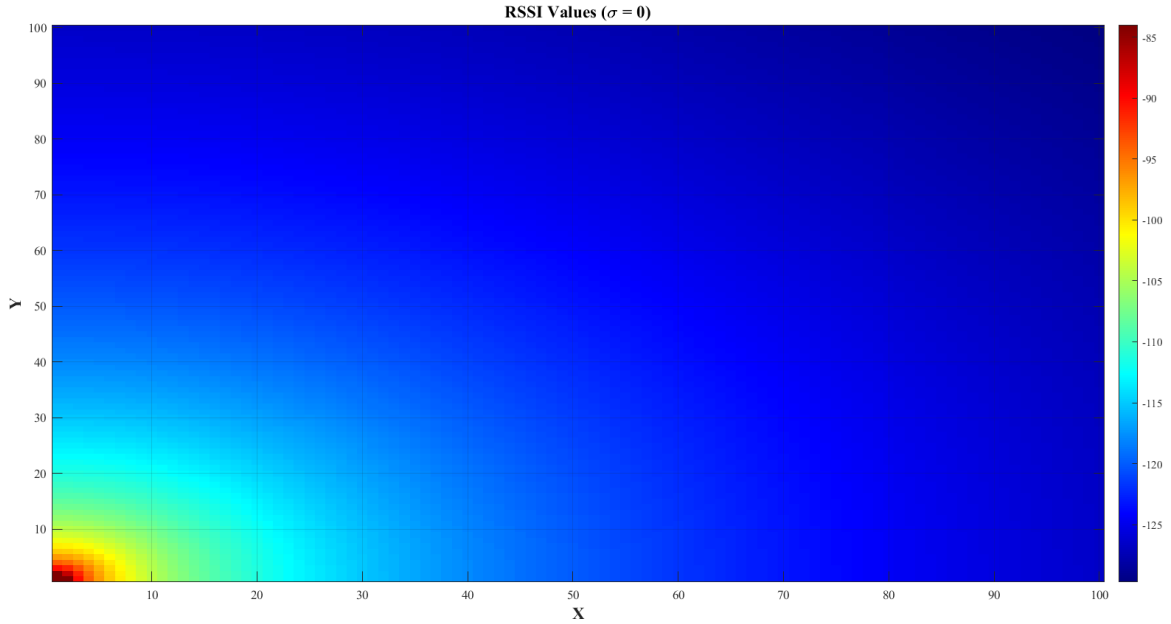

Figure 4.19: RSSI Value($\sigma = 0$)

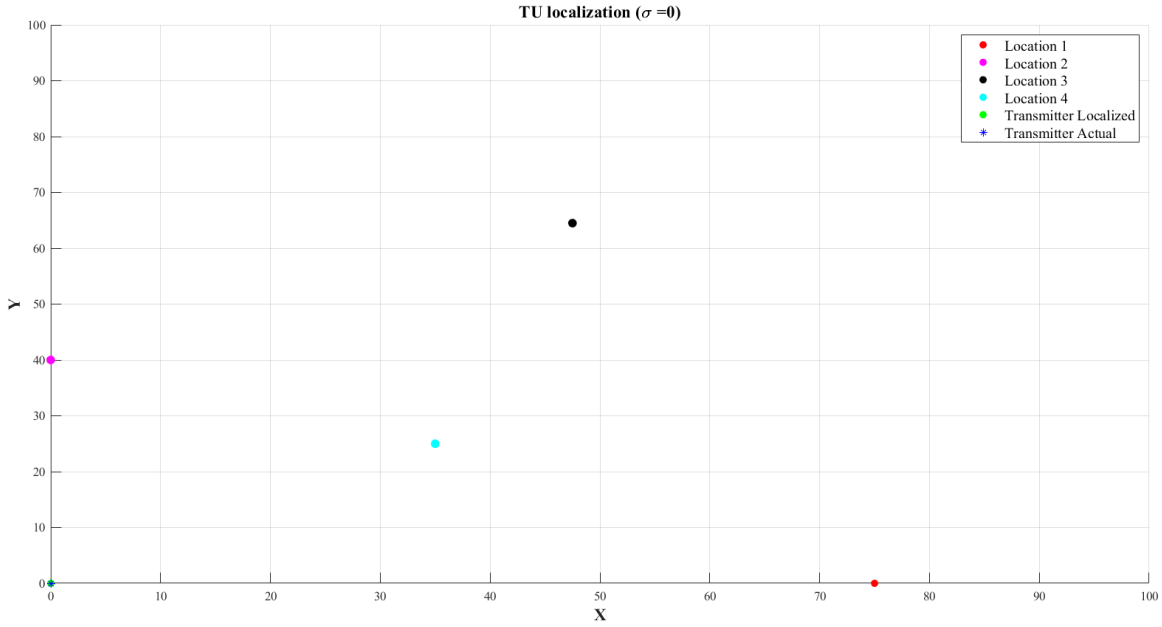

Figure 4.20: TU Localization($\sigma = 0$)

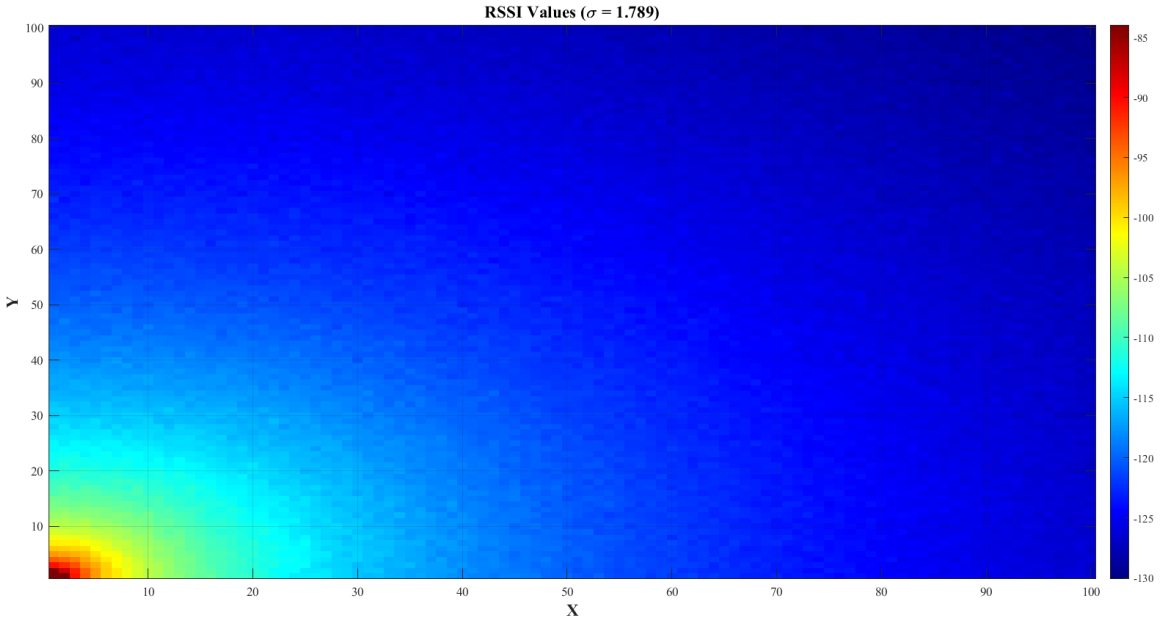

Figure 4.21: RSSI Value($\sigma = 1.789$)

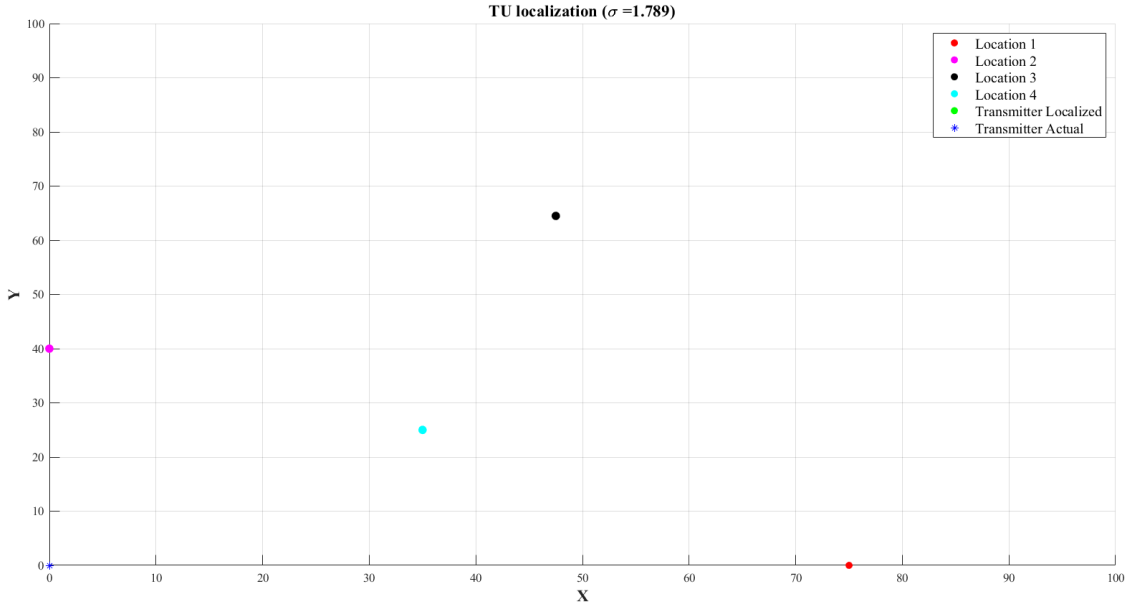

Figure 4.22: TU Localization($\sigma = 1.789$)

4.4. Mission Flight Results

4.4.1. First Mission Flight Data

From the experimental result, the RSSI values for each four predefined waypoints,average RSSI values,target coordinates are tabulated below:

Table 4.11: First mission flight data

Waypoints	RSSI(dBm)	Average RSSI(dBm)
1	-99,-99,-87,-84,-84,-85,-86,-83,-75,-74	-85.6
2	-68,-73,-74,-70,-74,-70,-74,-73,-75,-78	-72.9
3	-94,95,-95,-94,-95,-95,-94,-95,-96,-95	-94.8
4	-95,-92,-94,-88,-88,-87,-88,-87,-87,-87	-89.3

Table 4.12: First mission localized GPS coordinates

Mission	Localized GPS Coordinates	
	Latitude	Longitude
1	27.6834207	85.3218163

Table 4.13: First mission flight transmitter actual GPS coordinates

Mission	Transmitter Actual GPS Coordinate	
	Latitude	Longitude
1	27.6834307	85.3217547

Table 4.14: First mission flight data

Mission	Distance error(m)
1	6.16

According to the first mission's results, the transmitter's actual position is within the intended minimum distance error of 6.16 meters.If there had been no GPS error at the measuring sites, the outcomes might have been much better.

This,according to the log file, was displaying inaccuracies of one meter.The project's primary goal was effectively achieved with this mission flight test.

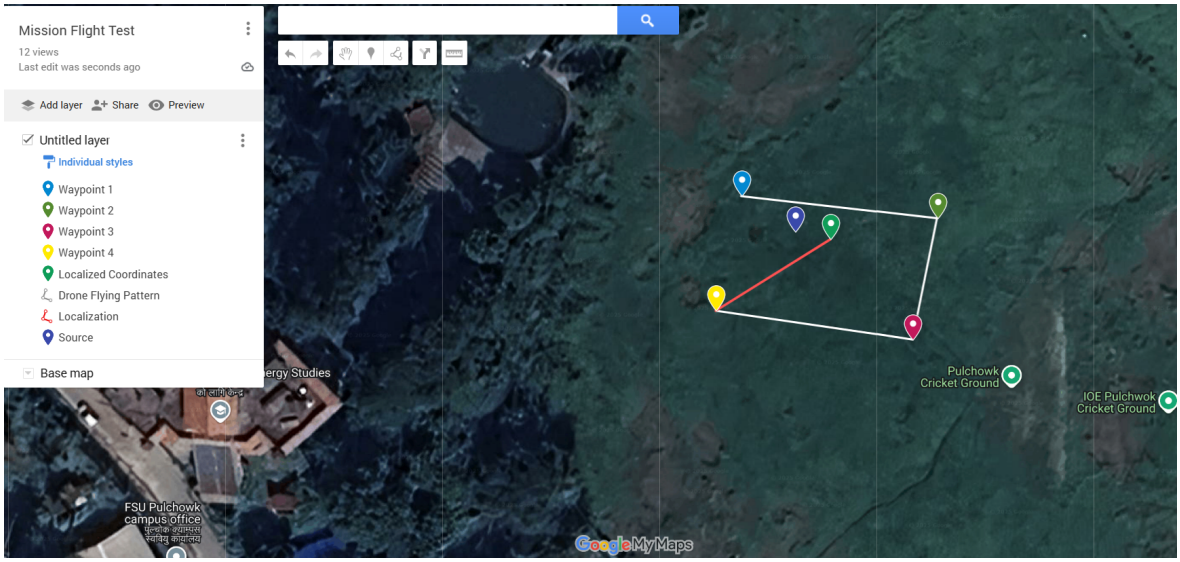

Figure 4.23: First mission flight at pulchowk campus

4.4.2. Second Mission Flight Data

From the experimental result, the RSSI values for each four predefined points,average RSSI values,target coordinates are tabulated below:

Table 4.15: Second mission flight data

Waypoints	RSSI(dBm)	Average RSSI(dBm)
1	-88,-88,-88,-88	-88.0
2	-88,-87,-81,-75	-82.75
3	-74,-73,-73,-73	-73.5
4	-74,-79,-79,-82	-78.5

Table 4.16: Second mission localized GPS coordinates

Mission	Localized GPS Coordinates	
	Latitude	Longitude
1	27.6834669	85.3217013

Table 4.17: Second mission transmitter actual GPS coordinates

Mission	Transmitter Actual GPS	
	Latitude	Longitude
2	27.6834898	85.3216944

According to the second mission's results,the transmitter's actual position is within the intended minimum distance error of 2.625 meters.The results may have been better if there

Table 4.18: Second mission flight data

Mission	Distance error(m)
1	2.625

were no GPS errors at measurement locations, which showed inaccuracies of 1 meter as seen in the log file.The mission flight test achieved the project's major objective.

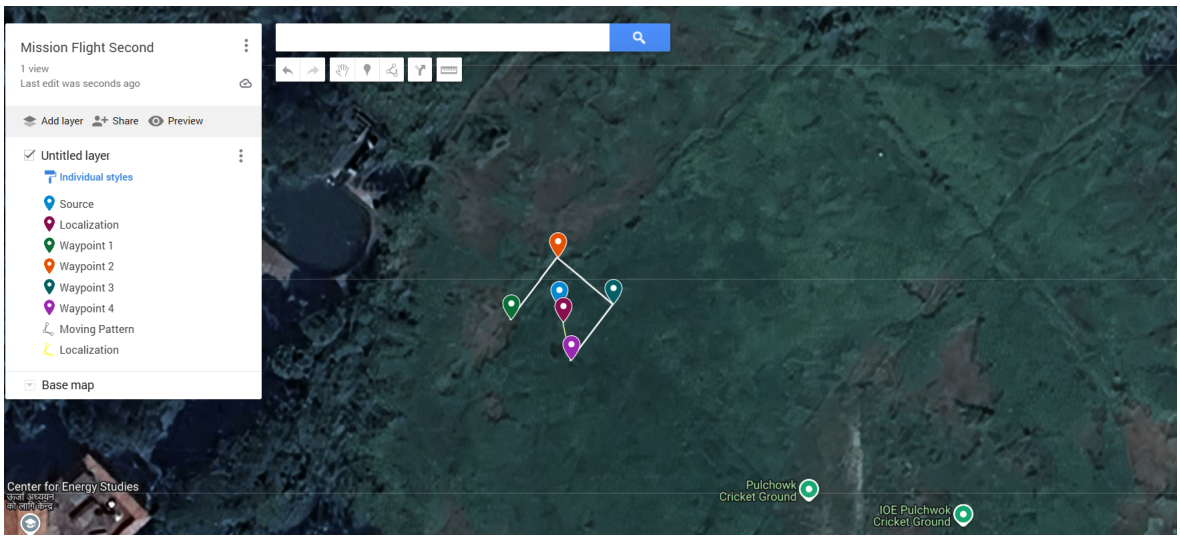

Figure 4.24: Second mission flight at pulchowk campus

4.4.3. Third Mission Flight Data

From the experimental result, the RSSI values for each four predefined waypoints,average RSSI values,target coordinates are tabulated below:

Table 4.19: Third mission flight data

Waypoints	RSSI(dBm)	Average RSSI(dBm)
1	-84, -88, -96, -85, -86, -85, -88, -89, -92, -88	-88.1
2	-84, -86, -85, -94, -93, -99, -94, -95, -91, -92	-91.3
3	-85, -92, -83, -90, -94, -93, -92, -90, -96, -100	-91.5
4	-96, -92, -100, -75, -88, -81, -83, -91, -87, -95	-88.8

Table 4.20: Third mission localized GPS coordinates

Mission	Localized GPS Coordinates	
	Latitude	Longitude
3	27.6884709	85.3218902

Table 4.21: Third mission transmitter actual GPS coordinates

Mission	Transmitter Actual GPS	
	Latitude	Longitude
3	27.6834616	85.3218892

Table 4.22: Third mission flight data

Mission	Distance error(m)
3	1

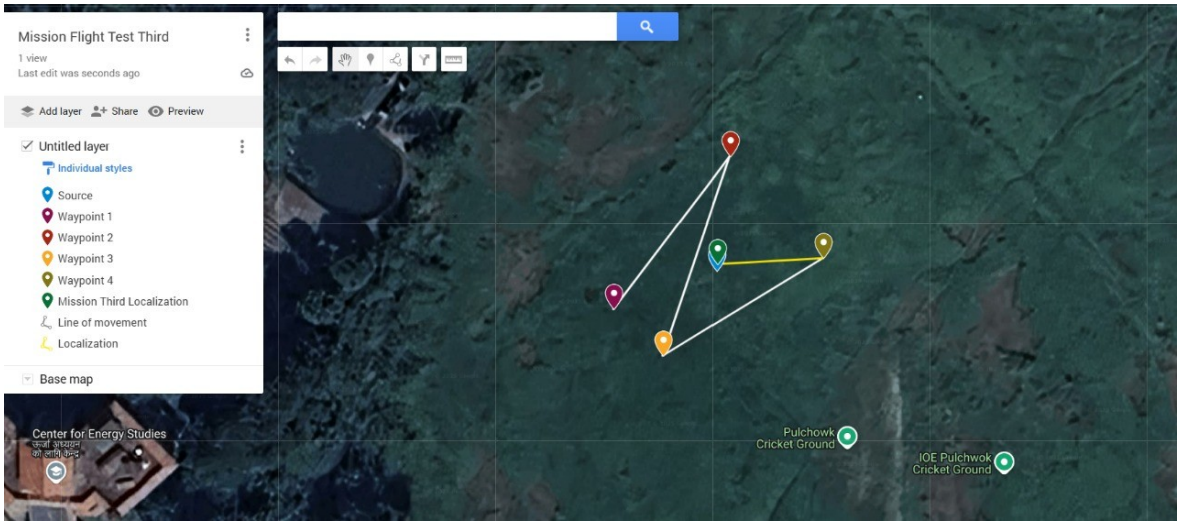

Figure 4.25: Third mission flight at Pulchowk Campus

The calculated distance between the two GPS points is about one meter.Thus,the RF source is localized with minimal distance inaccuracy.The main mission findings indicate that the transmitter is localized within the specified minimum distance error of one.

meter with regard to its current position.The results could have been better if there were no GPS inaccuracies at measurement locations, which showed errors of 1 meter.The mission test successfully achieves the project's major objective.

4.4.4. Fourth Mission Flight Data

From the experimental result, the RSSI values for each four predefined points,average RSSI values,estimated distances,target coordinates are tabulated below:

Table 4.23: Fourth mission flight data

Waypoints	RSSI(dBm)	Average RSSI(dBm)
1	-96, -96, -98, -87, -88, -87, -87, -84, -92, -98	-91.3
2	-94, -96, -97, -99, -96, -99, -96, -95, -96, -91	-95.9
3	-95, -91, -94, -96, -96, -93, -85, -85, -90, -91	-91.6
4	-96, -94, -98, -97, -93, -94, -97, -95, -97, -89	-95.0

Table 4.24: Fourth mission localized GPS coordinates

Mission	Localized GPS Coordinates	
	Latitude	Longitude
4	27.6834645	85.3218830

Table 4.25: Fourth mission transmitter actual GPS coordinates

Mission	Transmitter Actual GPS	
	Latitude	Longitude
4	27.6834616	85.3218892

Table 4.26: Fourth mission flight data

Mission	Distance error(cm)
4	70

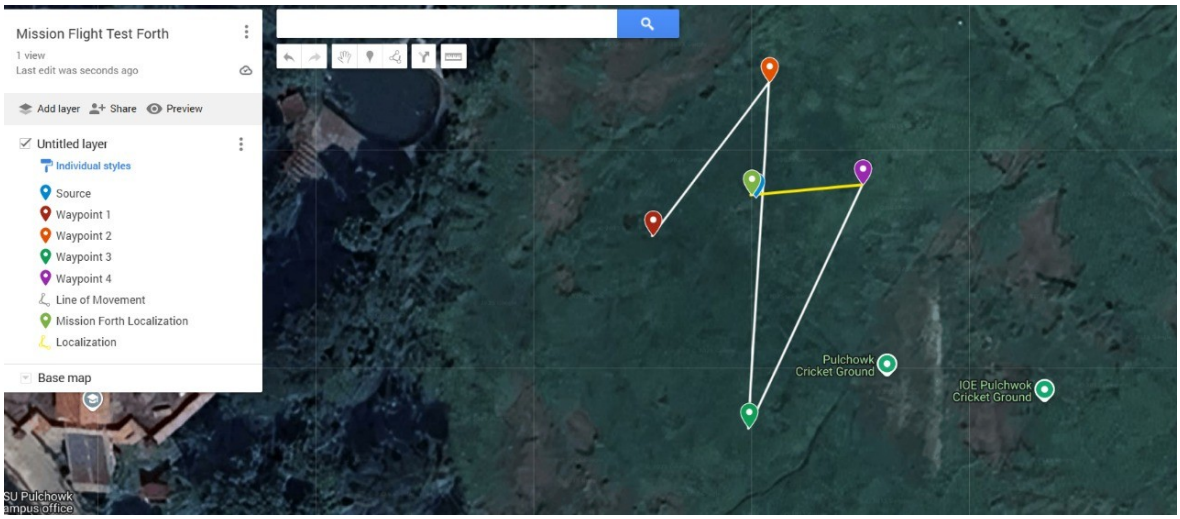

Figure 4.26: Fourth mission flight at Pulchowk Campus

The estimated distance between two GPS points is 70 centimeters.As a result,the RF Source is localized with the lowest possible distance error.Results from the main mission indicate that the transmitter is located within the specified minimum distance error of 70 centimeters

relative to its real position.The results could have been better if there wasn't a 1meter inaccuracy in GPS measurement places.The mission test successfully achieves the project's major objective.

4.5. Project Performance Benchmark: Current vs. Previous Analysis

4.5.1. Indoor Results

The experimental data for indoor localization, along with a comparison of indoor data and the accuracy of the current project, are listed in Table 4.27 and Table 4.28 below.

Table 4.27: Indoor Data Comparison

Distance (m)	Project	Distance Error (m)	Error/Distance
3.35	First Batch	1.29	0.385
4	Current	0.0749	0.018725

Table 4.28: Percentage (%) Accuracy of the Current Project

Compared to First Batch	94.19

4.5.2. Outdoor LoRa Results

The experimental data for outdoor localization, along with a comparison of indoor data and the accuracy of the current project, are listed in Table 4.29 and Table 4.30 below.

Table 4.29: Outdoor LoRa Data Comparison

Distance (m)	Project	Distance Error (m)	Error/Distance
80	First Batch	14	0.175
65	Current	0.42	0.0064615

Table 4.30: Percentage Accuracy (%) of Current Project

Compared to Previous First	96.30

4.5.3. Mission Flight Results

The experimental data for mission flight tests localization, along with a comparison of mission flight data and the accuracy of the current project, are listed in Table 4.31 and Table 4.32 below.

Table 4.31: Mission Flight Results

Flight Length (m)	Mission Flight No.	Distance (m)				Average Distance	Error (m)
		D1	D2	D3	D4		
81	1	10	22	25	17	18.5	6.16
35	2	8	8	9	10	8.75	2.625
92	3	18	17	17	17	17.25	1
135	4	17	18	36	17	22	0.7

Mean Error= 2.621m

Table 4.32: Previous Mission Flight Data

Project	Flight Length (m)	Mission Flight No.	Distance (m)	Error (m)
First Batch	50	1	50	25.6
Second Batch	38	1	38	7.32

Table 4.33: Comparison Table

Project	Error/Distance
First Batch	0.512
Second Batch	0.19263
Current	0.15765

Table 4.34: Percentage (%)Accuracy of the Current Project

Compared to	Percentage
First Batch	69.21%
Second Batch	18.16%

4.6. Limitations

1. The drone may frequently face GPS error, causing deviations from its intended waypoints.
2. The altitude given by M8N GPS module can fluctuate significantly leading to considerable altitude loss.
3. Multiple flight mission is not always feasible due to battery limitations.
4. The 2GB variant of the Raspberry Pi 4 may frequently experience lag during flight tests.
5. The Gaussian noise value used in simulation may or may not exactly match with the surrounding as noise is not constant.
6. It will only be able to determine the stationary source in two dimension.
7. The proposed localization modality is influenced by multi-path effects and signal fading.
8. The proposed system is based on the assumption that the transmitter emits with constant power.

4.7. Problem Faced

1. Difficult to model the exact noise in environment.Noise in the environment varies from time to time.
2. The RSSI value achieved cannot be fully noise proof due to the drone's own noise.

3. The readings of RSSI and localization were not collected at the very first glance due to connection issues, although that does not affect the localization.Connection can be lost between drone and ground terminal due to data issue.

4. Frequent GPS error were causing drone to deviate from its predefined path.M8N GPS module showed a lot of error particularly in uneven areas.

5. Strong wind disturbs in localization as it didn't allow drone to hover over a place.

6. Due to vibrations, the nuts in the frame become lose.

7. Raspberry Pi 4's operating system has a tendency to crash frequently due to software bugs.

4.8. Work Schedule

The work related to this project is divided into many sub tasks which is given its respective duration of completion.The scheduling of the work is tabulated below:

Table 4.35: Work Scheduling

Tasks	Duration
Research	8 months
Proposal Writing	11 days
Order of materials	7 months
Software Programming	7 months
Assembly	22 days
Testing of UAS	5 days
Mission Flight Testing	8 days
Mission data recording	5 days
Documentation	10 months
Final year report	30 days

4.8.1. Gantt chart

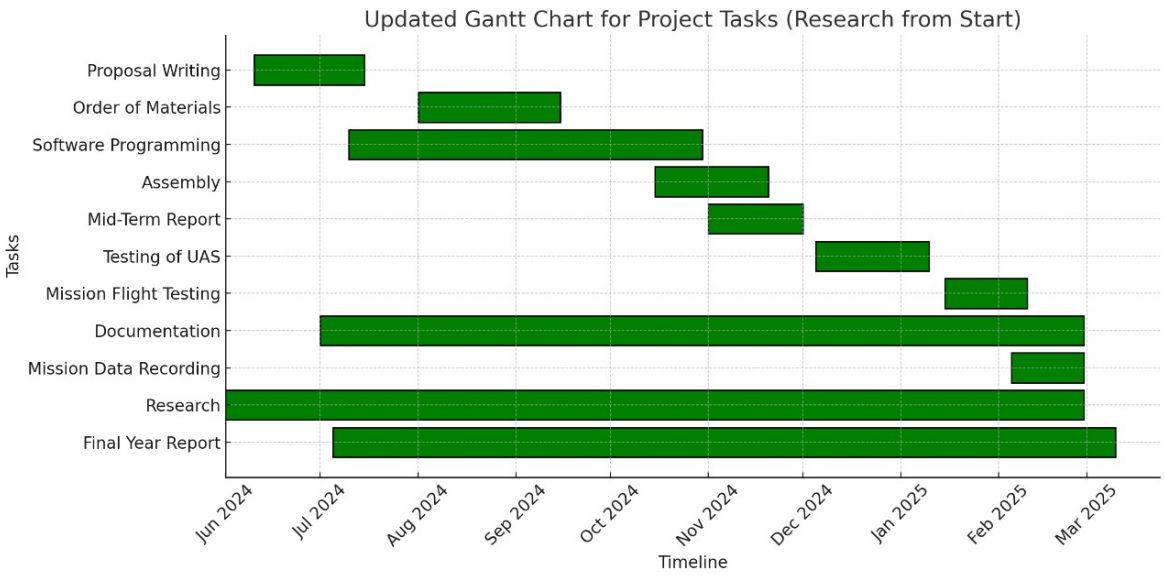

Figure 4.27: Gantt chart

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

In this research, we successfully designed an autonomous Unmanned Aerial System (UAS) (Quadcopter) for localizing RF sources localization, focusing on static targets and we also implemented both indoor and outdoor localization using the LoRa module and simulated the system in MATLAB.The results demonstrated the effectiveness of the LoRa technology for accurate localization, providing a low cost and energy efficient solution.The simulations validated the system's feasibility and laid a strong foundation for further development in real world applications.The system leverages the Received Signal Strength Indicator (RSSI) and multilateration approach to accurately localized the position of RF sources in both indoor and outdoor environments.The project demonstrated the feasibility of using low cost, energy efficient LoRa modules and WiFi based localization methods for precise target localization. Key achievements of the project include:

- Indoor Localization:The system achieved accurate localization of static RF sources in indoor environments, with a path loss exponent of 4.7275 and a minimal distance error of 0.0749 meters during testing.
- Outdoor Localization:In outdoor scenarios, the system demonstrated robust performance,with a path loss exponent of 2.85 and a distance error of 0.42 meters.The system's ability to adapt to environmental factors such as temperature and humidity further enhanced its reliability.
- Mission Flights:Four mission flights test were conducted, with the first achieving a distance error of 6.16m and the second reducing the error to 2.625m, third 1m and fourth is 70cm approximately.These results validate the system's capability to localize static targets with high precision, even in the presence of GPS error and environmental noise.
- Hardware and Software Integration:The integration of Raspberry Pi 4, Pixhawk, and LoRa modules enabled autonomous localization and real time data processing, making the system a practical solution for applications such as disaster rescue, environmental monitoring, and border surveillance.

Despite the challenges posed by GPS errors,environmental noise, and hardware limitations, the project successfully met its objectives.The system's accuracy and reliability were further improved through antenna calibration.

5.2. Future Enhancements

In the future, we aim to enhance this system by incorporating;

- Dynamic target localization to handle moving targets effectively.
- Additionally, we plan to integrate an obstacle avoidance system and path planning algorithms to enable autonomous navigation in complex environments.
- Localization through Multi UAV Collaboration.
- Furthermore, we plan to use directional antenna together with omnidirectional antenna.

CHAPTER 6: ADDITIONAL WORK

6.1. Dynamic Localization

The localization of a moving RF source in a real time is simply known as dynamic localization of RF source.Continuous or real time tracking and estimation of the position of a radio frequency (RF) emitter in environments where conditions are changing either because the source and/or receiver is moving, or the environment itself is dynamic.

6.1.1. Challenges in Dynamic Localization

1. **Multipath and NLOS Effects:**

   In dynamic environments, RF signals often encounter reflections and scattering from obstacles.These multi path and non line of sight (NLOS) conditions make it difficult to reliably extract direct path information,leading to significant localization errors.

2. **Doppler Shifts and Motion Dynamics:**

   When either the RF source or the receiver is moving, Doppler effects introduce frequency shifts.These shifts complicate the measurement of timing features such as time of arrival (TOA) and time difference of arrival (TDOA), thus affecting localization accuracy.

3. **Real Time Processing Constraints:**

   Dynamic scenarios require real time data processing.The need to rapidly process noisy and fluctuating RF measurements demands efficient, low-latency algorithms often a challenge given the computational complexity of techniques like particle filters or Kalman filters,etc.

4. **Sensor Synchronization and Calibration:**

   Accurate localization typically involves data from multiple sensors.In dynamic settings, precise synchronization and calibration are crucial, even small timing offsets or calibration errors can lead to large discrepancies in position estimates.

5. **Environmental Interference and Variability:**

   Outdoor and dynamic indoor environments are subject to rapidly changing conditions

and interference from other RF sources.Variability in factors such as signal attenuation, interference, and ambient noise can degrade measurement quality and thus hinder reliable localization.

6.2. LoRa Test at Campus

We have tried to do the dynamic target localization analysis.For this, site was pulchowk cricket ground.The receiver was fixed to a point and source moves forward in a given interval of time.It moved from time 0 second to 15 seconds with 5 second interval.The average velocity was calculated 0.4074 m/s.So, from the table 6.1 we can predict the position of the moving target in terms of distance only.To predict the direction we have to incorporate directional antenna as well.

Table 6.1: Dynamic localization data

RSSI Values	Actual Distance (m)	Calculated Distance (m)	Instantaneous Velocity (m/s)
-94	10	13.1920	0.39
-96	11.9693	16.50	0.3938
-99	13.93863	23.113	0.4523
-102	16.20	32.35	0.3938

Dynamic localization coordinate system and actual path is shown in the figure below:

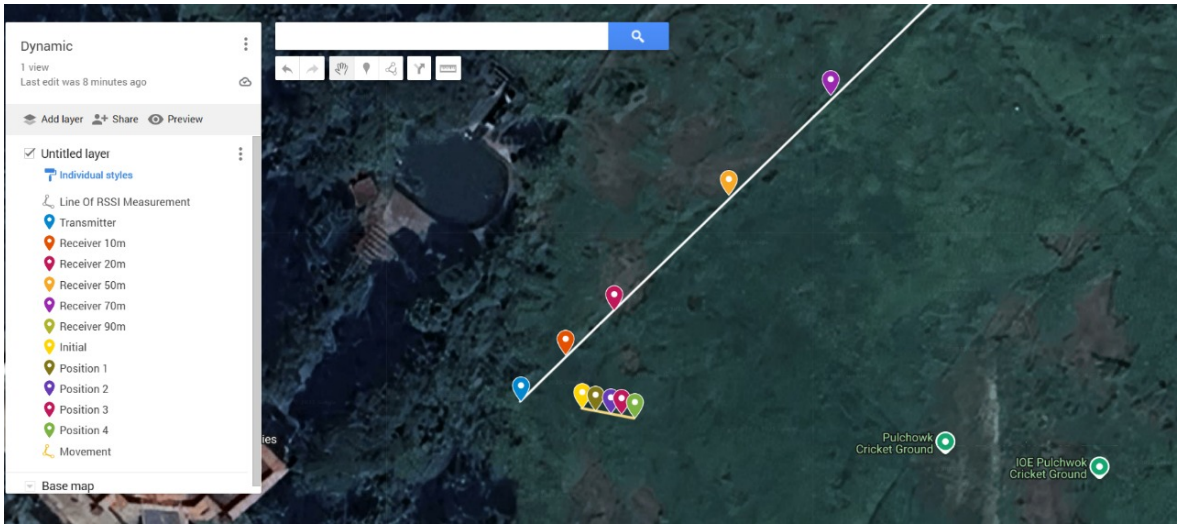

Figure 6.1: Dynamic localization data

References

[1] I. Jami, M. Ali, and R. Ormondroyd, "Comparison of methods of locating and tracking cellular mobiles," in *IEE Colloquium Novel Methods of Location and Tracking of Cellular Mobiles and their System Applications*. IET, 1999, pp. 1–1.

[2] H. Kwasme, "A study of rssi localization performance using lorawan and sdr," Master's thesis, Oklahoma State University, 2019.

[3] N. Güzey, "Rf source localization using multiple uavs through a novel geometrical rssi approach," *Drones*, vol. 6, no. 12, p. 417, 2022.

[4] S. Moro, V. Teeda, D. Scazzoli, L. Reggiani, and M. Magarini, "Experimental uav-aided rssi localization of a ground rf emitter in 865 mhz and 2.4 ghz bands," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–6.

[5] A. Ayala, L. Portela, F. Buarque, B. J. Fernandes, and F. Cruz, "Uav control in autonomous object-goal navigation: a systematic literature review," *Artificial Intelligence Review*, vol. 57, no. 5, p. 125, 2024.

[6] Y. Li, Y. Zhuang, X. Hu, Z. Gao, J. Hu, L. Chen, Z. He, L. Pei, K. Chen, M. Wang *et al.*, "Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4035–4062, 2020.

[7] M. L. Sichitiu, V. Ramadurai, and P. Peddabachagari, "Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements." in *International Conference on Wireless Networks*, vol. 2003, 2003, pp. 300–305.

[8] D. Oh and J. Han, "Smart search system of autonomous flight uavs for disaster rescue," *Sensors*, vol. 21, no. 20, p. 6810, 2021.

[9] S. Yucer, F. Tektas, M. V. Kilinc, I. Kandemir, H. Celebi, Y. Genc, and Y. S. Akgul, "Rssi-based outdoor localization with single unmanned aerial vehicle," *arXiv preprint arXiv:2004.10083*, 2020.

[10] M. I. M. Ismail, R. A. Dzyauddin, S. Samsul, N. A. Azmi, Y. Yamada, M. F. M. Yakub, and N. A. B. A. Salleh, "An rssi-based wireless sensor node localisation using trilateration and multilateration methods for outdoor environment," *arXiv preprint arXiv:1912.07801*, 2019.

[11] H. Kwon and I. Guvenc, "Rf signal source search and localization using an autonomous uav with predefined waypoints," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. IEEE, 2023, pp. 1–6.

[12] M. Barralet, X. Huang, and D. Sharma, "Effects of antenna polarization on rssi based location identification," in *2009 11th International Conference on Advanced Communication Technology*, vol. 1. IEEE, 2009, pp. 260–265.

[13] K. R. Tiwari, I. Singhal, and A. Mittal, "Real time rssi compensation for precise distance calculation using sensor fusion for smart wearables."

[14] L. K. Dressel, *Efficient and Low-cost Localization of Radio Sources with an Autonomous Drone*. Stanford University, 2018.

[15] M. Hasanzade, Ö. Herekoğlu, R. Yeniçeri, E. Koyuncu, and G. İnalhan, "Rf source localization using unmanned aerial vehicle with particle filter," in *2018 9th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2018, pp. 284–289.

[16] I. D. Olayanju and O. P. Ojelabi, "Using multilateration and extended kalman filter for localization of rfid passive tag in nlos," 2010.

[17] R. Maidana, A. Amory, and A. Salton, "Outdoor localization system with augmented state extended kalman filter and radio-frequency received signal strength," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 604–609.

[18] L. K. Dressel and M. J. Kochenderfer, "Efficient and low-cost localization of radio signals with a multirotor uav," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1845.

[19] A. Saravanakumar, T. Ayyasamy, and K. Senthilkumar, "Enhanced uav localization in gps-denied environments using acoustic tdoa and ekf integration," *Intelligent Service Robotics*, pp. 1–18, 2025.

[20] F. Koohifar, I. Guvenc, and M. L. Sichitiu, "Autonomous tracking of intermittent rf source using a uav swarm," *IEEE Access*, vol. 6, pp. 15 884–15 897, 2018.

[21] A. Wessels, X. Wang, R. Laur, and W. Lang, "Dynamic indoor localization using multilateration with rssi in wireless sensor networks for transport logistics," *Procedia Engineering*, vol. 5, pp. 220–223, 2010.

[22] P. Tong, X. Yang, Y. Yang, W. Liu, and P. Wu, "Multi-uav collaborative absolute vision positioning and navigation: A survey and discussion," *Drones*, vol. 7, no. 4, p. 261, 2023.

[23] S. Jayasekara, A. Al-Hourani, B. Ristic, and A. Skvortsov, "Autonomous uav search for an rf source in urban environments," in *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2020, pp. 1–6.

[24] F. Dou, J. Lu, T. Xu, C.-H. Huang, and J. Bi, "A bisection reinforcement learning approach to 3-d indoor localization," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6519–6535, 2020.

[25] M. Hasanzade, Ö. Herekoğlu, R. Yeniçeri, E. Koyuncu, and G. İnalhan, "Rf source localization using unmanned aerial vehicle with particle filter," in *2018 9th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2018, pp. 284–289.

[26] S. Brassai and I. Z. Székely, "Radio frequency (rf)-based, real-time, indoor localization system for unmanned aerial vehicles and mobile robot applications," *Acta Polytechnica Hungarica*, vol. 17, no. 9, 2020.

[27] H. V. Nguyen, M. Chesser, L. P. Koh, S. H. Rezatofighi, and D. C. Ranasinghe, "Trackerbots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals," *Journal of Field Robotics*, vol. 36, no. 3, pp. 617–635, 2019.

[28] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, and A. Sciarrone, "On the localization of wireless targets: A drone surveillance perspective," *IEEE Network*, vol. 35, no. 5, pp. 249–255, 2021.

[29] E. Pagliari, L. Davoli, and G. Ferrari, "Wi-fi-based real-time uav localization: A comparative analysis between rssi-based and ftm-based approaches," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[30] J. Siva and C. Poellabauer, "Robot and drone localization in gps-denied areas," *Mission-Oriented Sensor Networks and Systems: Art and Science: Volume 2: Advances*, pp. 597–631, 2019.

[31] Z. Shaikhanov, A. Boubrima, and E. W. Knightly, "Autonomous drone networks for sensing, localizing and approaching rf targets," in *2020 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2020, pp. 1–8.

[32] N. S. Ahmad, "Recent advances in wsn-based indoor localization: A systematic review of emerging technologies, methods, challenges and trends," *IEEE Access*, 2024.

[33] ——, "Recent advances in wsn-based indoor localization: A systematic review of emerging technologies, methods, challenges and trends," *IEEE Access*, 2024.

[34] M. Kato, T. K. Rodrigues, T. Abe, and T. Suganuma, "Exploiting radio frequency characteristics with a support unmanned aerial vehicle to improve wireless sensor location estimation accuracy," *IEEE Internet of Things Journal*, 2024.

[35] D. S. Jasrotia and M. J. Nene, "Localisation using uav in rfid and sensor network environment: Needs and challenges," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019, pp. 274–279.

[36] A. Alnoman, A. S. Khwaja, A. Anpalagan, and I. Woungang, "Emerging ai and 6g-based user localization technologies for emergencies and disasters," *IEEE Access*, 2024.

[37] J. Yan, H. Zhao, X. Luo, C. Chen, and X. Guan, "Rssi-based heading control for robust long-range aerial communication in uav networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1675–1689, 2018.

[38] H. V. Nguyen, M. Chesser, L. P. Koh, S. H. Rezatofighi, and D. C. Ranasinghe, "Trackerbots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals," *Journal of Field Robotics*, vol. 36, no. 3, pp. 617–635, 2019.

[39] S. Sharafeddine, "An optimized uav trajectory planning for localization," 2020.

[40] T. Aziz and I. Koo, "A comprehensive review of indoor localization techniques and applications in various sectors." *Applied Sciences (2076-3417)*, vol. 15, no. 3, 2025.

APPENDICES

CHAPTER A: Arduino Code for Receiver

A.1. Appendix

```
#include <LoRa.h>
#include <SPI.h>
#define SS_PIN 15
#define RST_PIN 16
#define DIO0_PIN 4

void setup() {
    Serial.begin(9600);
    while (!Serial);
    Serial.println("Receiver_Host");
    LoRa.setPins(SS_PIN, RST_PIN, DIO0_PIN);
    if (!LoRa.begin(433E6)) {
        Serial.println("LoRa_Error");
        while (1);
    }
    LoRa.receive();
}

void loop() {
    if (LoRa.parsePacket()) {
        Serial.print("RSSI: ");
        Serial.println(LoRa.packetRssi());
    }
}
```

Listing 1: Arduino Code for Receiver

APPENDICES

CHAPTER B: Arduino Code for Transmitter

B.1. Appendix

```
#include <ESP8266WiFi.h>
#include <LoRa.h>

#define SS 15   // LoRa radio's chip select pin
#define RST 16  // LoRa radio reset pin
#define DIO0 2  // Pin connected to DIO0 of LoRa radio

String data = "help!";  // Data to be sent

void setup() {
  Serial.begin(9600);
  while (!Serial);  // Wait for Serial to be ready
  Serial.println("Sender Host");

  LoRa.setPins(SS, RST, DIO0);

  if (!LoRa.begin(433E6)) { // Change the frequency to your desired
     frequency (433 MHz)
    Serial.println("LoRa Error");
    while (1);
  }
}

void loop() {
  Serial.print("Sending Data: ");
  Serial.println(data);

  LoRa.beginPacket();
  LoRa.print(data);
  LoRa.endPacket();

  delay(1000);  // Wait 2 seconds before sending the next packet
}
```

Listing 2: Arduino Code for Transmitter

APPENDICES

CHAPTER C: Matlab Code (RSSI Vs Distance PLot)

C.1. Appendix

```
% MATLAB code to generate RSSI vs Distance for different path loss exponents
clc;
clear;

% Parameters
d0 = 1; % Reference distance in meters
RSSI0 = -84; % Path loss at reference distance (dB)
d = linspace(1, 150, 500); % Distance values in meters

% Path loss exponents
n_values = 2.062;

% Initialize figure
figure;
hold on;

% Plot RSSI for each path loss exponent
for i = 1:length(n_values)
    n = n_values(i);
    RSSI = RSSI0 - 10 * n * log10(d / d0);
    plot(d, RSSI, 'LineWidth', 3, 'DisplayName', ['n = ' num2str(n)]);
    legend('FontSize', 12);

end

% Customize the plot
xlabel('Distance (m)', 'FontSize', 14);
ylabel('RSSI (dBm)', 'FontSize', 14);
legend('Location', 'northeast');
grid on;
xlim([0 150]);
ylim([-160 20]);
set(gca, 'YDir', 'reverse');
title('RSSI vs Distance for Different Path Loss Exponents', 'FontSize', 14);
```

```
35 hold off;
```

Listing 3: RSSI VS Distance Plot

APPENDICES

CHAPTER D: Matlab Code (Multilateration plot)

D.1. Appendix

```
1  % Define grid size and grid points
2  grid_size = 100;
3  step_size = 1;
4  [X,Y] = meshgrid(0:step_size:grid_size-step_size, 0:step_size:
       grid_size-step_size);
5
6  % Define transmitter location
7  tx_x = 2;
8  tx_y = 2;
9
10 % Define reference distance and reference RSSI
11 d0 = 2;
12 RSSI0 =-53.93;
13
14 % Define path loss exponent
15 n = 2.85;
16
17 % Define variance of the Gaussian noise
18 sigma = 3.6675;
19
20 % Calculate distance of each grid point from the transmitter
21 D = sqrt((X - tx_x).^2 + (Y - tx_y).^2);
22
23 % Calculate path loss
24 path_loss = 10 * n * log10(D / d0);
25
26 AWGN_mean = zeros(size(X));
27 for j = 1:1:50
28     Random = randn(size(X));
```

```
    AWGN_mean = AWGN_mean + Random;
end
AWGN_mean = sigma * AWGN_mean / 50;

% Calculate the RSSI
RSSI = RSSI0 - path_loss - AWGN_mean;

% Plot the RSSI values
figure();
imagesc(RSSI);
colormap('jet');
colorbar;
xlabel('X', 'FontWeight', 'bold','FontSize',14);
ylabel('Y', 'FontWeight', 'bold','FontSize',14);
set(gca, 'YDir', 'normal');
set(gca, 'fontname', 'times');
title('RSSI Values (\sigma = 3.6675)','FontSize',14);
grid on;

% Input four measurement locations
location1 = input('Enter coordinates for location 1 in format [X1 Y1]:
    ');
location2 = input('Enter coordinates for location 2 in format [X2 Y2]:
    ');
location3 = input('Enter coordinates for location 3 in format [X3 Y3]:
    ');
location4 = input('Enter coordinates for location 4 in format [X4 Y4]:
    ');

% Read the RSSI values at four measurement locations
RSSI1 = interp2(X, Y, RSSI, location1(1), location1(2));
RSSI2 = interp2(X, Y, RSSI, location2(1), location2(2));
RSSI3 = interp2(X, Y, RSSI, location3(1), location3(2));
RSSI4 = interp2(X, Y, RSSI, location4(1), location4(2));

if isnan(RSSI1) || isnan(RSSI2) || isnan(RSSI3) || isnan(RSSI4)
    disp('Invalid location entered');
else
    % Convert RSSI to distance using the path loss model
    d1 = d0 * 10^((RSSI0 - RSSI1) / (10 * n));
    d2 = d0 * 10^((RSSI0 - RSSI2) / (10 * n));
    d3 = d0 * 10^((RSSI0 - RSSI3) / (10 * n));
    d4 = d0 * 10^((RSSI0 - RSSI4) / (10 * n));
end

% Solving system of linear equations for four locations
```

```
x1 = location1(1);
y1 = location1(2);
x2 = location2(1);
y2 = location2(2);
x3 = location3(1);
y3 = location3(2);
x4 = location4(1);
y4 = location4(2);

% Define the system of equations (for four locations)
f = @(x) [((x(1) - x1)^2 + (x(2) - y1)^2 - d1^2);
          ((x(1) - x2)^2 + (x(2) - y2)^2 - d2^2);
          ((x(1) - x3)^2 + (x(2) - y3)^2 - d3^2);
          ((x(1) - x4)^2 + (x(2) - y4)^2 - d4^2)];

% Define the Jacobian matrix for four locations
J = @(x) [2 * (x(1) - x1), 2 * (x(2) - y1);
          2 * (x(1) - x2), 2 * (x(2) - y2);
          2 * (x(1) - x3), 2 * (x(2) - y3);
          2 * (x(1) - x4), 2 * (x(2) - y4)];

% Initial guess for the solution
x0 = [1; 1];

% Solve the system of equations using fsolve
options = optimoptions(@fsolve, 'Display', 'off');
x = fsolve(f,x0,options);

% Extract the solution
sol_x = x(1);
sol_y = x(2);
disp(['The localised x coordinate is: ',num2str(sol_x)])
disp(['The localised y coordinate is: ',num2str(sol_y)])

% Plot the result
figure();
scatter(location1(1), location1(2), 'r', 'filled');
hold on;
scatter(location2(1), location2(2), 50, 'm', 'filled');
scatter(location3(1), location3(2), 50, 'k', 'filled');
scatter(location4(1), location4(2), 50, 'c', 'filled');
scatter(sol_x, sol_y, 'g', 'filled');
scatter(0, 0, '*', 'b');  % Actual transmitter location
legend('Location 1', 'Location 2', 'Location 3', 'Location 4', '
   Transmitter Localized', 'Transmitter Actual', 'FontSize', 12);
xlim([0 grid_size]);
```

```
ylim([0 grid_size]);
set(gca, 'YDir', 'normal');
set(gca, 'fontname', 'times');
xlabel('X', 'FontWeight', 'bold', 'FontSize', 14);
ylabel('Y', 'FontWeight', 'bold', 'FontSize', 14);
title('Indoor localization (\sigma =3.6675)', 'FontSize', 14);
grid on;[]
```

Listing 4: Multilateration plot

APPENDICES

CHAPTER E: Simple Moving Average filter

E.1. Appendix

```
1
2  def moving_average(data, window_size):
3      """
4      Calculate the simple moving average (SMA) of the provided data.
5
6      Parameters:
7        data (list or iterable): The RSSI readings.
8        window_size (int): The number of data points to average.
9
10     Returns:
11       list: A list of the SMA filtered values.
12     """
13     if window_size <= 0:
14         raise ValueError("Window size must be positive")
15
16     averages = []
17     for i in range(len(data)):
18         # Use a smaller window for the initial elements
19         if i < window_size - 1:
20             window = data[0:i+1]
21         else:
22             window = data[i-window_size+1:i+1]
23         averages.append(sum(window) / len(window))
```

```
    return averages

# Example usage:
rssi_readings = [-70, -68, -75, -69, -72, -71, -70, -67]
window_size = 3
filtered_rssi = moving_average(rssi_readings, window_size)
print("Filtered RSSI (pure Python):", filtered_rssi)
```

Listing 5: Simple Moving Average filter

APPENDICES

CHAPTER F: Python code for mission flight

F.1. Appendix

```
import asyncio
import serial
import time
import os
import subprocess
from mavsdk import System
import numpy as np

# Earth radius in meters (WGS84)
EARTH_RADIUS = 6371000  # in meters

def is_serial_port_free(port):
    """Check if the serial port is currently being used by another
       process."""
    try:
        result = subprocess.check_output(['sudo', 'lsof', port])
        if result:
            print(f"Port {port} is currently being used. You need to
               free it before proceeding.")
            return False
        return True
    except subprocess.CalledProcessError:
        return True
```

```
22
23 def open_serial_port(port, baudrate, timeout=5, retries=5):
24     """Open the serial port, retrying on failure."""
25     for _ in range(retries):
26         try:
27             # Check if the port is free before trying to open it
28             if not is_serial_port_free(port):
29                 print(f"Retrying to open {port}...")
30                 time.sleep(1)
31                 continue
32
33             ser = serial.Serial(port, baudrate, timeout=timeout)
34             print(f"Successfully opened {port}")
35             return ser
36         except serial.SerialException as e:
37             print(f"Failed to open {port}: {e}. Retrying...")
38             time.sleep(1)
39     raise serial.SerialException(f"Could not open {port} after {
           retries} retries")
40
41 def omni(ser_acm0):
42     """Function to handle RSSI values from the serial input."""
43     def try_decode(serial_instance):
44         try:
45             return serial_instance.readline().decode('utf-8', errors='
                   replace').strip()
46         except UnicodeDecodeError:
47             return "UnicodeDecodeError"
48
49     try:
50         num_readings = 0
51         rssi_sum = 0
52
53         while num_readings < 10:
54             try:
55                 line_ACM0 = try_decode(ser_acm0)
56                 if line_ACM0.startswith("RSSI:_"):
57                     rssi_value_acm0 = int(line_ACM0.split("_")[1].
                           strip())
58                     print("RSSI_0 =", rssi_value_acm0)
59
60                     num_readings += 1
61                     rssi_sum += rssi_value_acm0
62             except KeyboardInterrupt:
63                 print("Exiting...")
64                 break
```

```
65
66          if num_readings > 0:
67              RSSI_REF = -77  # Renamed variable for clarity
68              n = 2.053 # Path loss exponent
69              average_rssi = rssi_sum / num_readings
70              print("Average RSSI:", average_rssi)
71
72              distance = round(10 ** ((RSSI_REF - average_rssi) / (10 *
                    n)), 2)
73              print("Estimated Distance:", distance)
74              return distance
75          else:
76              return None
77      except Exception as e:
78          print(f"Error in omni: {e}")
79          return None
80
81  def gps_to_cartesian(lat0, lon0, lat, lon):
82      """Convert GPS coordinates to Cartesian coordinates."""
83      lat0_rad = np.radians(lat0)
84      lon0_rad = np.radians(lon0)
85      lat_rad = np.radians(lat)
86      lon_rad = np.radians(lon)
87
88      dlat = lat_rad - lat0_rad
89      dlon = lon_rad - lon0_rad
90
91      x = dlon * EARTH_RADIUS * np.cos((lat0_rad + lat_rad) / 2)
92      y = dlat * EARTH_RADIUS
93
94      return x, y
95
96  def cartesian_to_gps(lat0, lon0, x, y):
97      """Convert Cartesian (x, y) back to latitude and longitude."""
98      lat0_rad = np.radians(lat0)
99
100     lat = np.degrees(lat0_rad + (y / EARTH_RADIUS))
101     lon = np.degrees(np.radians(lon0) + (x / (EARTH_RADIUS * np.cos(
            lat0_rad))))
102
103     return lat, lon
104
105 def find_target_coordinates(cartesian_coordinates, distances):
106     """Estimate target coordinates using trilateration."""
107     if len(cartesian_coordinates) < 4 or len(distances) < 4:
108         print("Not enough data points to solve for (x, y)")
```

```
109         return None
110
111     (x1, y1), (x2, y2), (x3, y3), (x4, y4) = cartesian_coordinates
112     d1, d2, d3, d4 = distances
113
114     A = np.array([
115         [x2 - x1, y2 - y1],
116         [x3 - x2, y3 - y2],
117         [x4 - x3, y4 - y3]
118     ])
119
120     B = np.array([
121         [(-d2**2 + d1**2 + x2**2 - x1**2 + y2**2 - y1**2) / 2],
122         [(-d3**2 + d2**2 + x3**2 - x2**2 + y3**2 - y2**2) / 2],
123         [(-d4**2 + d3**2 + x4**2 - x3**2 + y4**2 - y3**2) / 2]
124     ])
125
126     target_coords, _, _, _ = np.linalg.lstsq(A, B, rcond=None)
127     x_target, y_target = target_coords.flatten()
128     print(f"Target coordinates found: x={x_target}, y={y_target}")
129
130     return x_target, y_target
131
132 async def main():
133     """Main function to control the drone and handle distance
            calculation."""
134     try:
135         # Open serial port
136         ser_acm0 = open_serial_port('/dev/ttyACM0', 9600)
137         ser_acm0.reset_input_buffer()  # Clear old data
138
139         drone = System()
140         await drone.connect(system_address="serial:///dev/ttyAMA0
                :921600")
141
142         async for state in drone.core.connection_state():
143             if state.is_connected:
144                 print("Drone connected!")
145                 break
146
147         await drone.action.arm()
148         print("Drone armed")
149         await asyncio.sleep(2)
150
151         await drone.action.takeoff()
152         print("Taking off")
```

```
153         await asyncio.sleep(10)
154
155         waypoints = [
156             (27.6834888, 85.3216788, 1296),
157             (27.6834594, 85.3219899, 1296),
158             (27.6832882, 85.3219416, 1296),
159             (27.6833283, 85.3216395, 1296),
160         ]
161
162         lat0, lon0 = waypoints[0][0], waypoints[0][1]
163         altitude = waypoints[0][2]
164         distances = []
165         cartesian_coordinates = []
166
167         for i, (lat, lon, alt) in enumerate(waypoints, 1):
168             print(f"Moving to waypoint {i}")
169             await drone.action.goto_location(lat, lon, alt, 0.0)
170             await asyncio.sleep(35)
171
172             x, y = gps_to_cartesian(lat0, lon0, lat, lon)
173             cartesian_coordinates.append((x, y))
174
175             distance = omni(ser_acm0)  # Pass serial instance
176             if distance is not None:
177                 distances.append(distance)
178
179         print("Distances:", distances)
180         print("Coordinates:", cartesian_coordinates)
181
182         target_coords = find_target_coordinates(cartesian_coordinates,
                distances)
183         if target_coords:
184             x, y = target_coords
185             lat, lon = cartesian_to_gps(lat0, lon0, x, y)
186             print(f"Target GPS: {lat}, {lon}")
187             await drone.action.goto_location(lat, lon, altitude, 0.0)
188             await asyncio.sleep(25)
189
190         await drone.action.return_to_launch()
191         print("Landing...")
192     except Exception as e:
193         print(f"Error in main: {e}")
194     finally:
195         # Properly close the serial port to avoid blocking
196         if 'ser_acm0' in locals() and ser_acm0.is_open:
197             ser_acm0.close()
```

```
198             print("Serial port closed")
199
200 # Run the main function
201 asyncio.run(main())
```

Listing 6: Python code for mission flight

APPENDICES

CHAPTER G: Python code for distances calculation between two GPS Coordinates

G.1. Appendix

```
1  import math
2
3  def haversine_distance(lat1, lon1, lat2, lon2):
4      """
5      Calculate the great-circle distance between two points on the
           Earth using the Haversine formula.
6
7      Parameters:
8          lat1, lon1: Latitude and Longitude of point 1 (in decimal
               degrees)
9          lat2, lon2: Latitude and Longitude of point 2 (in decimal
               degrees)
10
11     Returns:
12         Distance in kilometers between the two points.
13     """
14     # Convert latitude and longitude from degrees to radians
15     lat1, lon1, lat2, lon2 = map(math.radians, [lat1, lon1, lat2, lon2
           ])
16
17     # Compute differences
18     dlon = lon2 - lon1
19     dlat = lat2 - lat1
20
21     # Haversine formula
```

```
    a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math
        .sin(dlon / 2)**2
    c = 2 * math.asin(math.sqrt(a))

    # Radius of Earth in kilometers (use 3956 for miles)
    r = 6371
    return c * r

# Example usage:
if __name__ == "__main__":
    lat1, lon1 = 27.6834616, 85.3218892
    lat2, lon2 = 27.683470923307837, 85.32189021073688
    distance = haversine_distance(lat1, lon1, lat2, lon2)
    print("Distance:", distance, "km")
```

Listing 7: Python code for distance calculation between two gps coordinates

CHAPTER H: GPS Uncertainty of second Mission flight

H.1. Appendix

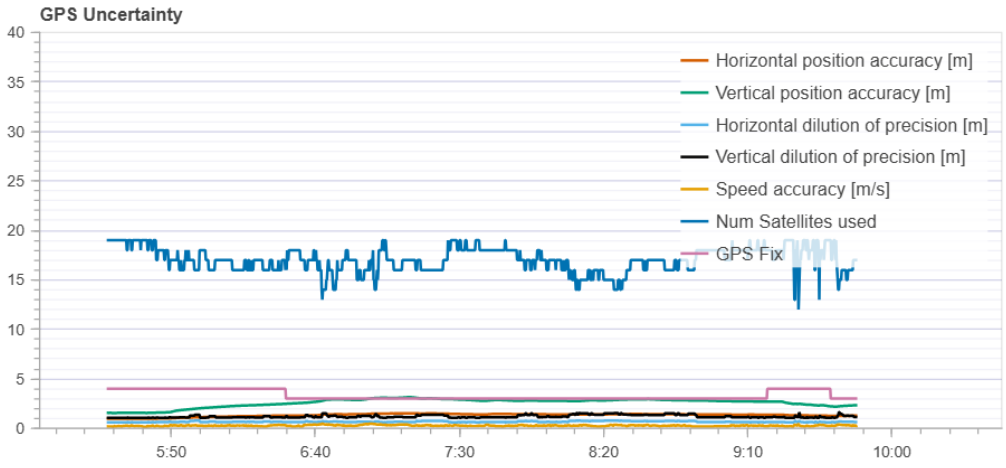

Figure H.1: GPS Uncertainty of Second Mission Flight

CHAPTER I: GPS Path Second Mission Flight

I.1. Appendix

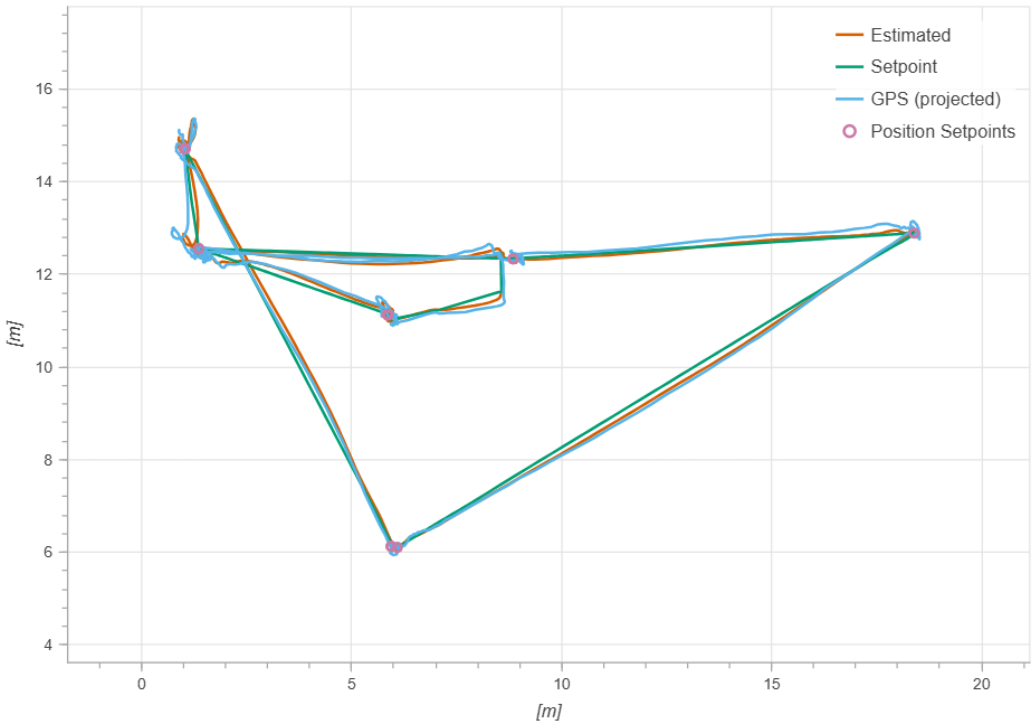

Figure I.1: GPS path of Second Mission Flight

CHAPTER J: Calculation of Distance between Two GPS Coordinates Using Cartesian Conversion(Mission Flight first)

J.1. Appendix

Calculation of Distance between Two GPS Coordinates Using Cartesian Conversion

Coordinates:

- Point A: Latitude = 27.683420766447526°, Longitude = 85.32181633019137°
- Point B: Latitude = 27.6834307°, Longitude = 85.3217547°

Assumptions: Earth's radius, $R = 6,371,000$ m

Step 1: Convert Degrees to Radians

$$\phi_1 = 27.683420766447526 \times \left(\frac{\pi}{180}\right) \approx 0.4829952\,\text{rad}$$
$$\lambda_1 = 85.32181633019137 \times \left(\frac{\pi}{180}\right) \approx 1.48915\,\text{rad}$$
$$\phi_2 = 27.6834307 \times \left(\frac{\pi}{180}\right) \approx 0.4829954\,\text{rad}$$
$$\lambda_2 = 85.3217547 \times \left(\frac{\pi}{180}\right) \approx 1.48915\,\text{rad}$$

Step 2: Compute the Differences in Radians

$$\Delta\phi = \phi_2 - \phi_1 \approx 1.733 \times 10^{-7}\,\text{rad}$$
$$\Delta\lambda = \lambda_2 - \lambda_1 \approx -1.076 \times 10^{-6}\,\text{rad}$$

Step 3: Convert to Cartesian Coordinates

Conversion formulas:

$$x = R \cdot \cos(\phi) \cdot \cos(\lambda), \quad y = R \cdot \cos(\phi) \cdot \sin(\lambda), \quad z = R \cdot \sin(\phi)$$

For small differences, we approximate the changes $(\Delta x, \Delta y, \Delta z)$ using partial derivatives:

$$\begin{aligned}
\Delta x &\approx -R\left[\sin(\phi_1)\cos(\lambda_1)\right]\Delta\phi - R\left[\cos(\phi_1)\sin(\lambda_1)\right]\Delta\lambda \\
\Delta y &\approx -R\left[\sin(\phi_1)\sin(\lambda_1)\right]\Delta\phi + R\left[\cos(\phi_1)\cos(\lambda_1)\right]\Delta\lambda \\
\Delta z &\approx R\left[\cos(\phi_1)\right]\Delta\phi
\end{aligned}$$

Using approximate trigonometric values at Point A:

$$\sin(\phi_1) \approx 0.464, \quad \cos(\phi_1) \approx 0.885, \quad \cos(\lambda_1) \approx 0.086, \quad \sin(\lambda_1) \approx 0.996$$

Compute Δx:

$$\begin{aligned}
\text{1. First term:} \quad & -R \cdot \sin(\phi_1) \cdot \cos(\lambda_1) \cdot \Delta\phi \\
&= -6,371,000 \cdot (0.464 \times 0.086) \cdot 1.733 \times 10^{-7} \approx -0.0441\,\text{m} \\
\text{2. Second term:} \quad & -R \cdot \cos(\phi_1) \cdot \sin(\lambda_1) \cdot \Delta\lambda \\
&= 6,371,000 \cdot (0.885 \times 0.996) \cdot 1.076 \times 10^{-6} \approx 6.034\,\text{m} \\
\Delta x &\approx -0.0441 + 6.034 \approx 5.990\,\text{m}
\end{aligned}$$

Compute Δy:

$$\begin{aligned}
&1.\,\text{First term:} \quad -R \cdot \sin(\phi_1) \cdot \sin(\lambda_1) \cdot \Delta\phi \\
&\qquad = -6,371,000 \cdot (0.464 \times 0.996) \cdot 1.733 \times 10^{-7} \approx -0.510\,\text{m} \\
&2.\,\text{Second term:} \quad R \cdot \cos(\phi_1) \cdot \cos(\lambda_1) \cdot \Delta\lambda \\
&\qquad \approx -0.520\,\text{m} \\
&\Delta y \approx -0.510 - 0.520 \approx -1.030\,\text{m}
\end{aligned}$$

Compute Δz:

$$\begin{aligned}
\Delta z &\approx R \cdot \cos(\phi_1) \cdot \Delta\phi \\
&= 6,371,000 \cdot 0.885 \cdot 1.733 \times 10^{-7} \approx 0.978\,\text{m}
\end{aligned}$$

Step 4: Compute the 3D Euclidean Distance

The distance, d, is given by:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

Calculate:

$$(\Delta x)^2 \approx (5.990)^2 \approx 35.88, \quad (\Delta y)^2 \approx (-1.030)^2 \approx 1.061, \quad (\Delta z)^2 \approx (0.978)^2 \approx 0.956$$

Thus,

$$d \approx \sqrt{35.88 + 1.061 + 0.956} \approx \sqrt{37.897} \approx 6.16\,\text{m}$$

Final Result: The estimated distance between the two GPS coordinates is approximately **6.16** meters.

CHAPTER K: QR code to access the mission flight video

K.1. Appendix

Figure K.1: QR code to access the mission flight video