Px4 sitl with gazebo failsafe

BACHELOR OF TECHNOLOGY IN COMPUTERSCIENCE & ENGINEERING

(ARTIFICIAL INTELLIGENCE& MACHINE LEARNING)

Submitted By G. Hemanth Kumar

Under the guidance of

Avijith Ashe

(Teaching Assitstant)

Department of computer science

(Approved by AICTE & Permanently Affiliated to JNTUK,KAKINADA and accredited by NAAC) Yanam Road,Korangi, Andhra Pradesh,pin-533461

Table of Contents

1. Introduction
   - Overview of the project
   - Importance of GPS failsafe for quadrotors
2. Objectives
   - Goals of the project
   - Expected outcomes
3. Feasibility Study
   - Availability of simulation tools
   - Justification for using PX4 SITL and Gazebo Classic
4. Timeline & Task Breakdown
   - Week 1: Research and study of necessary tools
   - Week 2: Setup of PX4 SITL and Gazebo Classic
   - Week 3: Implementation of GPS failsafe mechanism
   - Week 4: Testing, analysis, and refinement
5. Resource Estimation
   - Software requirements
   - Hardware specifications
6. Implementation Steps
   - Installation of required tools (PX4, ROS 2, Gazebo Classic, MAVLink, QGroundControl)
   - Setting up the simulation environment
   - Executing GPS failure scenarios
   - Data logging and analysis
7. Simulation Execution and Testing
   - Running the PX4 quadrotor in Gazebo
   - Inducing and recovering from GPS failure
   - Logging data using QGroundControl

8. Failsafe Analysis & Evaluation
    - Drone behavior before and after GPS failure
    - Position and altitude changes
    - Velocity and attitude variations
    - 3D trajectory analysis
9. Comparison Between Normal and GPS Failure Scenarios
    - Differences in movement patterns
    - Stability comparison
    - Performance assessment
10. Conclusion
    - Summary of findings
    - Effectiveness of the implemented failsafe mechanism
    - Future work and improvements

Abstraction

This project focuses on designing, implementing, and testing a GPS failure failsafe mechanism for a PX4-based quadrotor using PX4 SITL, Gazebo Classic, and ROS 2. The study aims to simulate real-world GPS failure scenarios and develop reliable recovery strategies to ensure quadrotor stability and safety. The implementation follows a structured approach, including setting up the simulation environment, executing GPS failure tests, and analyzing flight data. Through a comparative study of normal operation versus failure scenarios, the project evaluates the effectiveness of the failsafe system. The final findings will help enhance quadrotor performance in GPS-denied environments, contributing to the development of robust autonomous aerial systems.

PX4 Simulation Project Documentation Proposal (Week 1)

1. Introduction

This project focuses on designing and testing a GPS failure failsafe mechanism for a quadrotor. Using PX4 SITL, Gazebo Classic, and ROS 2, the simulation replicates real-world GPS failure scenarios to develop reliable recovery strategies.

2. Objectives

- Develop a failsafe system to detect and respond to GPS failure in a PX4-based quadrotor.
- Simulate GPS failure conditions within PX4 SITL and Gazebo Classic.
- Implement flight control strategies to maintain stability during GPS loss.
- Analyze flight data and telemetry to evaluate the effectiveness of the failsafe mechanism.

3. Feasibility

The project is feasible as PX4 SITL, Gazebo Classic, and ROS 2 offer a robust simulation environment, eliminating the need for physical hardware. Established GPS failure management techniques will be implemented and tested to validate the failsafe system.

4. Timeline

Week	Task
Week	Research and study necessary tools
Week	Set up the simulation environment
Week	Implement the failsafe mechanism in SITL
Week	Test, analyze, and refine the system

5. Resource Estimation

- **Software:** PX4 Autopilot, Gazebo Classic (Gazebo 11), ROS 2, QGroundControl
- **Hardware:** A high-performance Ubuntu-based computer for simulation

6. Conclusion

This proposal outlines a structured plan to develop a GPS failure failsafe system for a PX4-based quadrotor. With a well-defined timeline and resource allocation, the project aims to enhance quadrotor performance in GPS-denied environments through systematic simulation and testing.

Proposal selection and initial setup (Week2)

This project aims to design and simulate a failsafe GPS failure for a quadrotor employing PX4 SITL (Software-In-The-Loop) with Gazebo. It entails setting up the PX4 environment, source-building PX4, performing simulations, and viewing the quadrotor's behavior in a GPS failure simulation scenario. Learners will adhere to official PX4 tutorials and instructions to achieve a hands-on grasp of autonomous system simulation and failsafe concepts.

Steps and Tasks:

1. Setup and Installation:

- **Install ROS and Gazebo:**
- Install ROS 2

sudo apt update

sudo apt install ros-humble-desktop

Install the compatible version of Gazebo.

    sudo apt install gazebo11 libgazebo11-dev

- **Install MAVLink:**
- Install MAVLink libraries to facilitate communication between PX4 and Gazebo.

    sudo apt install python3-pip

    pip3 install pymavlink mavproxy

- **Build PX4 from Source:**

- Clone the PX4 repository from GitHub.

  git clone https://github.com/PX4/PX4-Autopilot.git cd
  PX4-Autopilot

  git submodule update --init --recursive

- Run the build command: make px4_sitl gazebo to set up the Gazebo simulation environment.

  make px4_sitl gazebo

- **Install QGroundControl (QGC):**

- Download the latest AppImage from the official site.

  wget
  https://d176td9ibo4jno.cloudfront.net/latest/QGroundControl.AppImage

  chmod +x QGroundControl.AppImage

- And then launch the qground control

  ./QGroundControl.AppImage

GPS Failure Failsafe Implementation and Simulation for Quadrotor (Week 3)

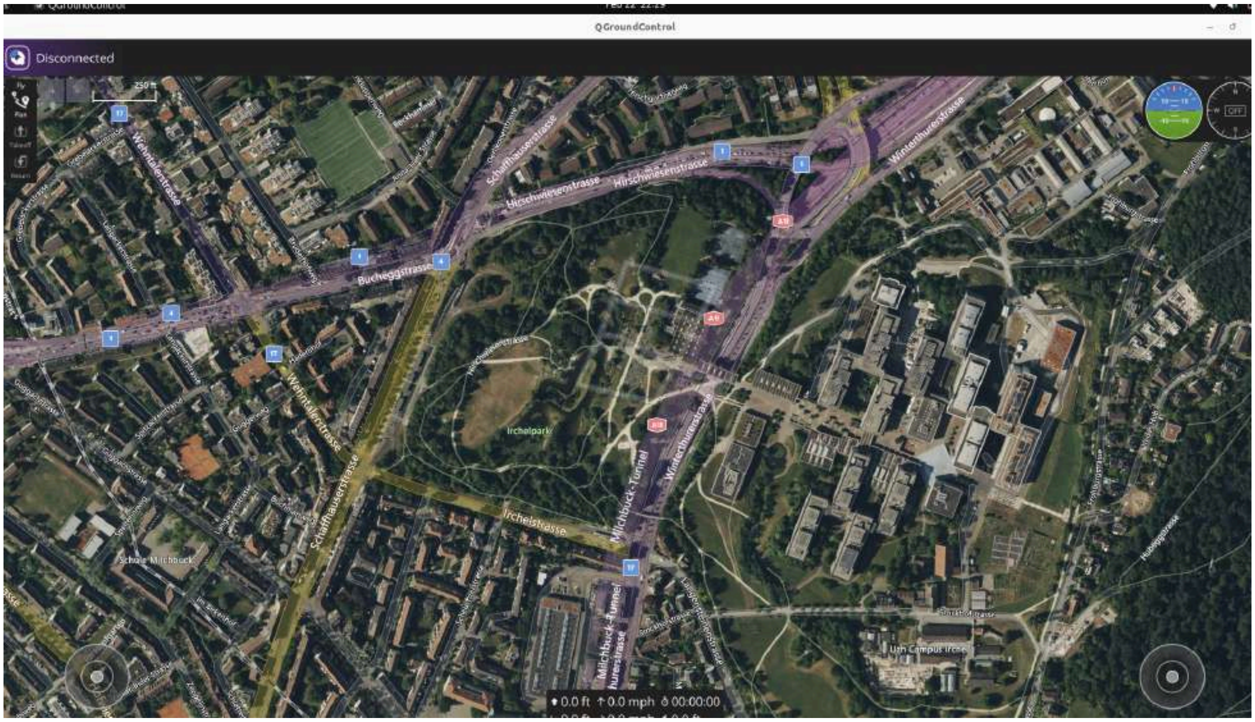
Week 2: Simulation Execution and Initial Tests

Step 1: Launch QGroundControl

Open a terminal and enter the following command:

```
./QGroundControl.AppImage
```

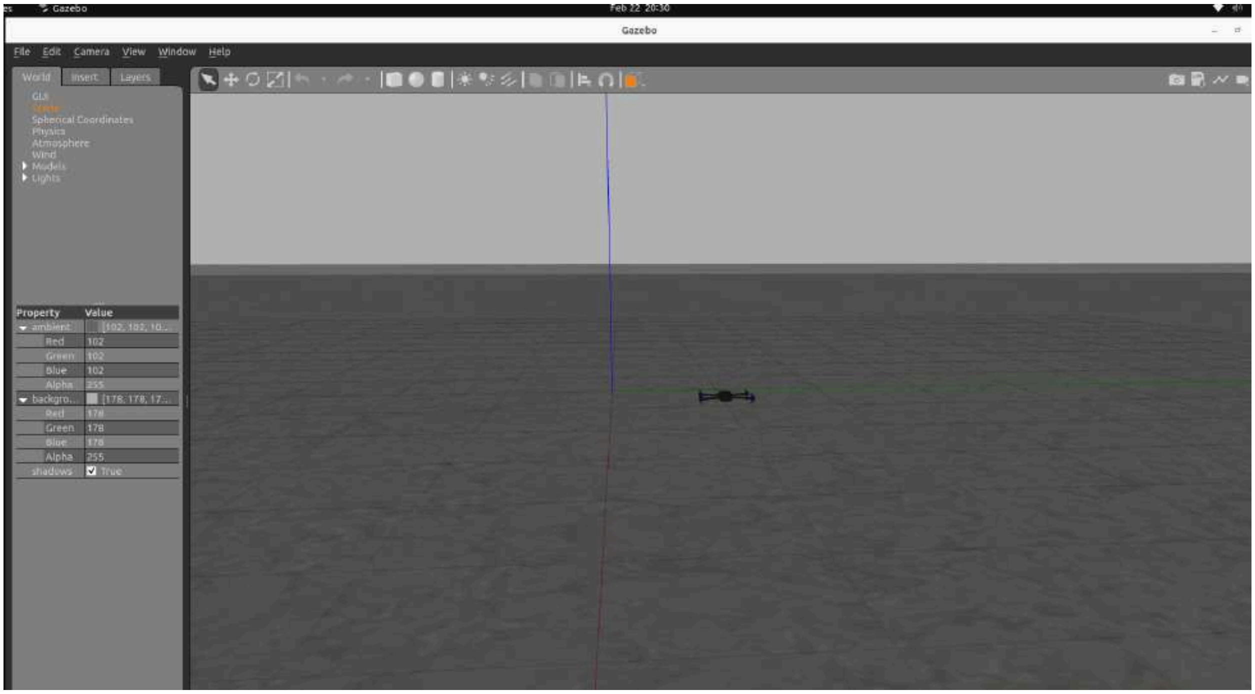
This will launch the QGroundControl GUI.



Step 2: Launch the PX4 Drone in Gazebo

Open a terminal and enter:

cd ~/px4-Autopilot
make px4_sitl gazebo



After running these commands, the Gazebo GUI will appear.

Go to the terminal where Gazebo launched :

Enter the command to make the drone stay at a position after a GPS failure :

param set COM_POSCTL_NAVL 0

param save

Shutdown

And restart the gazebo by using the above commands now make the gps failsafe

Step 3: Take the Drone

You can take off the drone using one of the following methods:

1. **Using QGroundControl:**
    - Use the available options in the QGroundControl GUI to launch the drone.

OR

2. **By using the Terminal:**

Return to the terminal where you launched the PX4 drone and enter:
commander takeoff

- This command will make the drone take off and start flying.

Step 4: Set a Target Location

Once the drone takes off, assign a location to reach.

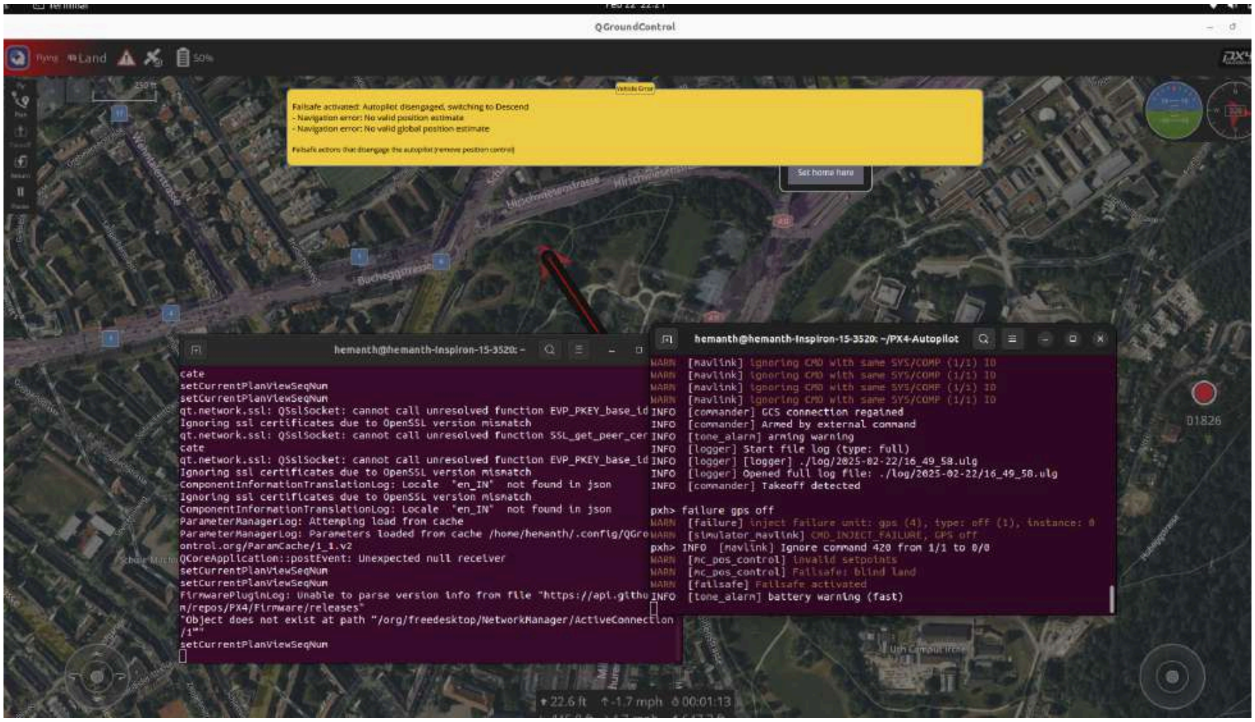
Step 5: Inducing GPS Failure

To simulate a GPS failsafe:

1. Open the **MAVLink Console** in QGroundControl:
   - Navigate to **Analyze Tools > MAVLink Console**.

Enter the following command to disable GPS:
failure gps off

2. This will cause GPS failure, and a corresponding message will be displayed.



Step 6: Restoring GPS Functionality

- After a few seconds, enter the following command to restore GPS:

failure gps ok

- This will stabilize the drone or allow it to continue to its destination.

After that, download the log files :

Analyze tools > logdownload > choose log file

Convert into a CSV file by using commands

Open the terminal :

Navigate to the folder where the ulog file is located:

Enter the command :   Ulog2csv file_path

GPS Failure Failsafe Implementation :

Final report :

Monitor and log data during the simulation using MAVLink or QGroundControl

How to download log data in QGROUNDCONTROL :

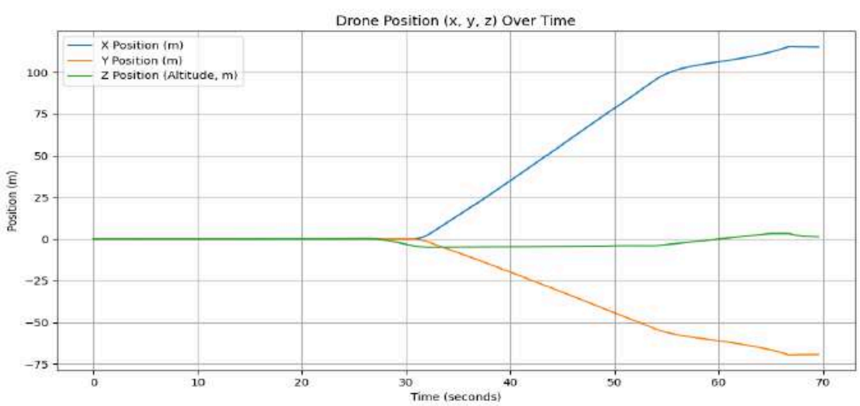
– Open QGroundControl by entering the command :

- ./QGroundControl.AppImage
- After that, click on the Q button a
- And then it shows three options
- And then click on analyze tools > log download
- Select and click on download

Evaluate how the failsafe mechanism ensures safe operation under GPS failure scenarios.

1. **Position and altitude**

Plot x, y, z position over time to observe the quadrator's behaviour post GPS failure :



Drone Behavior Before and After GPS Failure

Before GPS Failure (0 - 100 seconds)

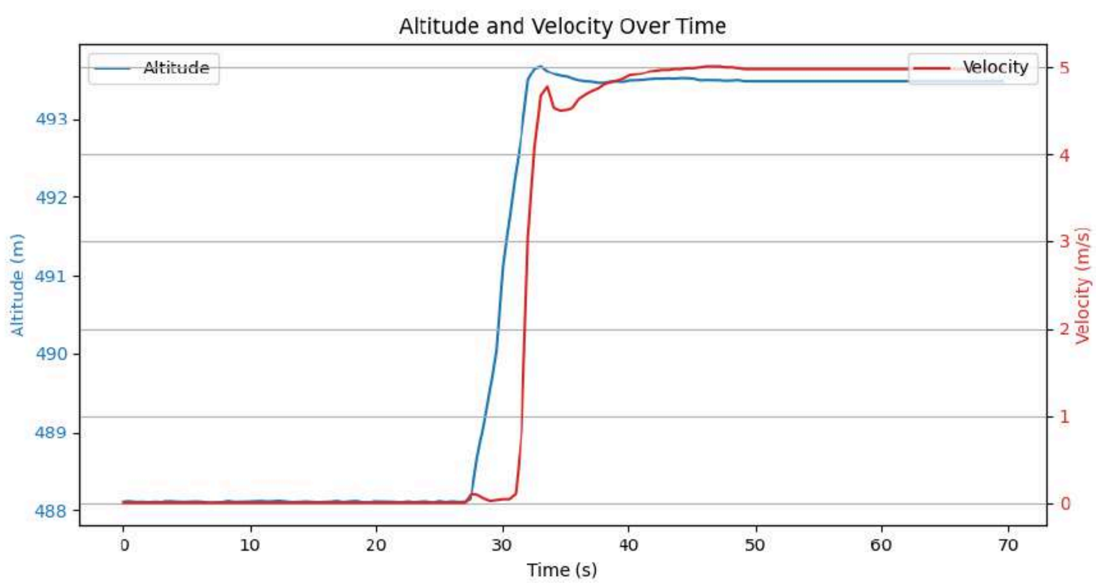
The quadrotor was **holding steady**, maintaining its position without noticeable movement. Everything seemed **normal**, with no unexpected drifts or instability.

After GPS Failure (Around 100 seconds)

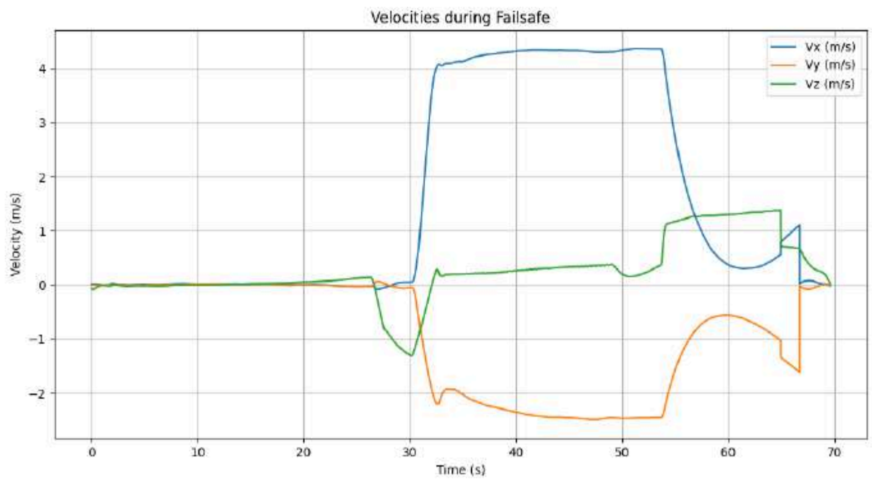
- X Position (Blue Line):
  The drone started drifting forward (+X direction) on its own. After a while, it slowed down but never fully stopped.
- Y Position (Orange Line):
  It suddenly veered sideways to the left (-Y direction), covering a significant distance before gradually correcting itself and returning closer to its original path.
- Z Position (Green Line, Altitude):
  The altitude remained fairly stable, with only minor ups and downs.

2. Failsafe activation.

Time-series plot of failsafe status by hovering at a position.



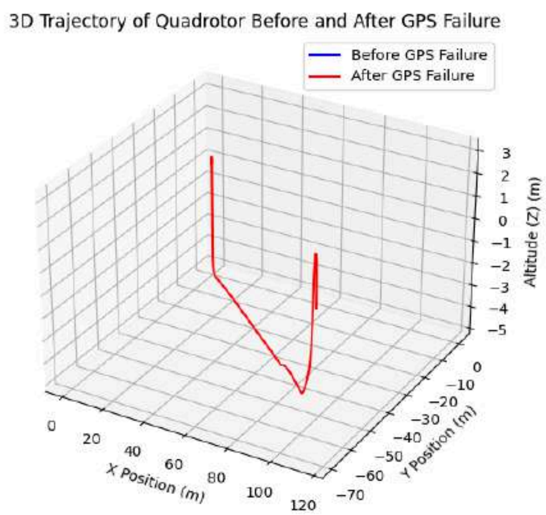
3. Velocity and Attitude :



- Stable Start (0-25s): The drone was stable with minimal movement.
- Failsafe Trigger (25-30s): A sudden speed increase (Vx spikes to 4.5 m/s) and instability in vertical (Vz) and sideways (Vy) movement.
- Unstable Phase (30-55s): The drone struggled to stabilize, with oscillations in all directions.
- Recovery (After 55s): Velocities gradually returned to normal, but small fluctuations suggest minor disturbances.

4. Trajectory

3d trajectory of the quadrotor in Gazebo before and after GPS failure.



5. Comparision :

Compare normal operation (without failure) and GPS failure scenarios to highlight differences.

Normal Operation (Without Failure)

- The drone moves smoothly and follows a planned path.
- It stays at a steady height without sudden drops.
- Speed and direction are stable.
- The position is tracked accurately.

GPS Failure Scenario

- The drone moves unpredictably and may drift.
- It changes height suddenly, losing stability.
- It doesn’t follow the planned path properly.
- The system tries to correct the movement, but it's not smooth.