ISRO Drone Navigation System – Project Documentation

Objective:

To develop an autonomous aerial vehicle (AAV) capable of:

- Navigating unknown indoor/outdoor arenas using computer vision
- Performing 3D mapping of the environment
- Detecting obstacles and identifying safe landing zones
- Executing Return-to-Home (RTH) algorithms autonomously

This solution was built as part of the ISRO IROC-25 practicum project and simulates real-world deployment scenarios for satellite maintenance and search-and-rescue missions.

Key Features:

1. **Autonomous Flight Navigation:**
    - Implemented using ROS2 (Robot Operating System 2)
    - Custom navigation stack for wayfinding without GPS
    - Path planning using Dijkstra/A* algorithms
2. **3D Mapping:**
    - Utilizes the OAK-D Lite camera for real-time depth perception
    - Generates point cloud data and 3D maps using OpenCV + depth APIs
    - Simulated environments built and tested using Gazebo on macOS
3. **Obstacle Detection & Avoidance:**
    - LiDAR simulation using Gazebo plugins

- Dynamic obstacle tracking and safe rerouting
- Real-time depth-based filtering

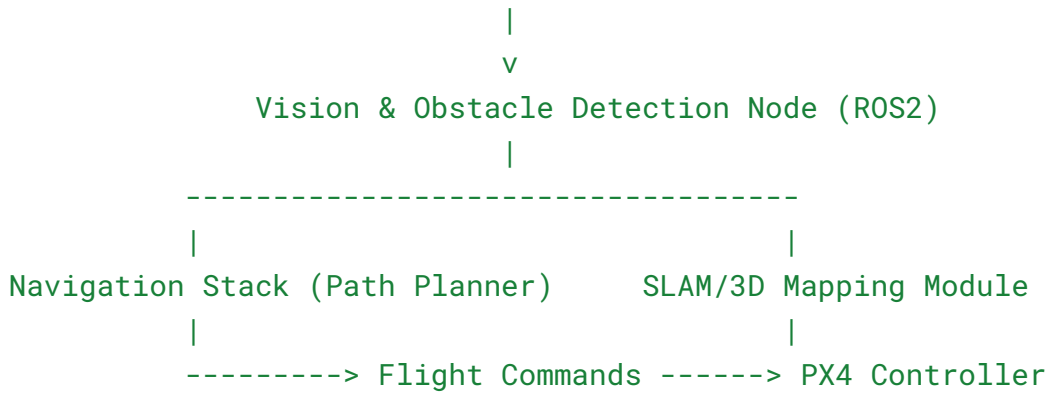
4. **Landing Zone Detection:**
    - Vision-based analysis using HSV and morphological filters
    - Identifies flat, obstacle-free surfaces
    - Autonomous descent using barometric/vision-based altimetry
5. **Return-to-Home (RTH):**
    - Failsafe RTH logic triggers on:
        - Battery threshold
        - Signal loss
        - Manual override
    - Uses breadcrumb path memory + obstacle-aware reverse routing

---

Tech Stack:

Category	Tool/Library
OS & Platform	macOS Catalina + ROS2 Humble
Simulation	Gazebo (Fortress), RViz2
Vision & Perception	OpenCV, DepthAI API (OAK-D Lite)
Languages	Python, XML (ROS launch files)
Drone Framework	PX4-compatible flight stack

System Architecture:
plaintext
CopyEdit
```
Drone Hardware (Simulated) --> OAK-D Lite (Depth Feed)
```



Challenges Faced:

- Running ROS2 and Gazebo efficiently on macOS (needed special patches and workarounds)
- Synchronizing real-time vision with ROS2 nodes due to USB camera delays
- Creating custom ROS2 nodes to handle return path memory without GPS

Future Improvements:

- Integrating GPS/IMU fusion for outdoor navigation
- Deploying the model on an actual drone with onboard edge compute (e.g., NVIDIA Jetson Nano)
- Real-time map export to cloud dashboards