Reflection and Justification

Project: UAV Strategic Deconfliction System
Author: Prathmesh Velhal
Course / Assignment: FlytBase Robotics Assignment 2025

---

1. Introduction

The primary objective of this project is to implement a **strategic UAV deconfliction system** that ensures safe operation of multiple drones in shared airspace.
The system checks for **spatial and temporal conflicts** between a primary drone mission and other active drone missions.

This document reflects the **design decisions, reasoning, and justification** behind the implementation.

---

2. Design Decisions

2.1 Data Structure

- **Waypoint:** A dictionary containing x, y, z, and time.
  - Justification: Provides flexibility to represent 3D position and time simultaneously.
- **Mission:** A dictionary containing drone_id and a list of waypoints.
  - Justification: Allows easy expansion for multiple drones and their trajectories.

2.2 Conflict Detection

- **Method:** 3D Euclidean distance between waypoints of different drones.
- **Parameters:**
  - safe_distance (meters) → minimum allowable separation between drones.
  - time_window (seconds) → maximum time difference to consider two positions simultaneous.
- **Justification:**
  - Euclidean distance ensures accurate spatial calculation in 3D space.

    - Temporal check prevents false positives when drones pass the same point at different times.

2.3 Visualization

- Plotted trajectories using **Matplotlib 3D plots**.
- Conflicts are marked with red X.
- **Justification:** Visual feedback makes it easier to validate and debug drone paths.

---

3. Reflections on Implementation

3.1 Strengths

- Modular and reusable code: Functions like detect_conflicts and plot_missions can be extended for more drones or larger missions.
- Clear visual representation of conflicts, helping operators make quick decisions.
- Lightweight and dependency-minimal (only requires matplotlib).

3.2 Limitations

- Assumes linear interpolation between waypoints; real drone motion may vary slightly.
- Conflict detection is simplified; does not account for drone size, velocity, or wind conditions.
- Real-time integration with drones or FlytBase SDK requires additional interfaces.

3.3 Future Improvements

- Integrate with FlytBase SDK for **live mission monitoring**.
- Add **dynamic safety radius** based on drone speed or payload.
- Implement **predictive conflict detection** using trajectory forecasting.

---

4. Justification of Design Choices

- Using **Python** ensures rapid development and integration with common drone simulation libraries.

- Folder structure (src/, docs/, visuals/) follows best practices for readability and submission clarity.
- Separation of **code, documentation, and visuals** ensures maintainability and easier evaluation.

Overall, the project is designed to be **robust, clear, and extensible**, providing a solid foundation for UAV mission deconfliction and further development in real-world scenarios.

---

5. Conclusion

This project successfully demonstrates the **strategic deconfliction of UAVs** in shared airspace using Python.
It balances **accuracy, simplicity, and clarity**, making it suitable for academic evaluation and potential real-world testing.