FlytBase assignment for AI robotics Intern

Name: Harshada Deshingkar

Email: harshadadeshingkar@yahoo.in, happiest.hd@gmail.com

Contact: +91 8888162255, +91 8888013819

Project Overview

This project implements a UAV/drone deconfliction system that verifies whether the primary drone's planned waypoint mission can be safely executed in a shared airspace. The system identifies spatial and temporal conflicts with simulated drone trajectories and resolves them using time delays.

Functionality

1. Interpolates each drone's 3D path over time.
2. The primary drone visits all the waypoints and reaches the final goal/ final waypoint.
3. Detects spatial and temporal conflicts using a configurable safety buffer.
4. Resolves conflicts by computing the minimal time delay for the primary drone (assuming the shared airspace cannot be altered and the spatiotemporal paths of other company drones cannot be changed so we delay our drone in order to avoid clash/conflict).
5. Visualizes both the original conflicting scenario and the conflict-resolved scenario as animated GIFs.

Modularity & Architectural Design

The system for deconfliction is designed with modularity and transparency. Fundamental tasks are divided into four modules:

- models.py: Establishes data structure for drones and waypoints to ensure uniform and expandable representation.
- conflict_detector.py: Interpolation, conflict detection, and conflict resolution logic are all performed in this module, enabling clean separation of spatial-temporal reasoning.
- simulator.py: Renders paths of drones and conflict points as animated GIFs, facilitating both debugging and presentation.
- main.py: Offers a test entry point and convenient customization of drone scenarios.

This organization supports scalability, easier debugging, and future extension potential (e.g., file-based inputs, integration with live APIs).

Spatial & Temporal Conflict Checks

Spatial conflicts are identified based on **Euclidean distance between drones** at shared or overlapping time intervals. The steps include:

- **Interpolation**: The waypoints of all drones are interpolated over time by a constant time step (dt), transforming sparse waypoints into smooth flight paths.
- **Temporal Overlap**: Overlap of mission time windows among the lead drone and other drones is checked by the system.
- **Distance Calculation**: 3D distance is calculated for every overlapping time point. Whenever the distance goes below a specified safety_buffer, it's labeled as a conflict.
- **Conflict Resolution**: The launch of the main drone is incrementally postponed until all conflicts are resolved (assuming the shared airspace cannot be altered and the spatiotemporal paths of other company drones cannot be changed), or until no feasible delay exists within the mission time frame.

Testing Strategy & Edge Cases

The following test scenarios were designed to evaluate the robustness of the deconfliction system across varied spatial and temporal situations:

1. High-traffic airspace congestion
2. Point collision between two drones
3. Partial path overlap
4. Multiple conflicts at different times
5. Edge cases: no possible conflict-free launch time possible
6. If multiple/all drones in shared airspace can be controlled then we can apply the same functionality to all the drones, find the minimum delay and execute that order of flight.
7. Modified case: If skipping waypoints is allowed to reach the goal faster, the drone can adjust its path to follow a minimal trajectory, maintaining the required safety buffer and direct towards the goal.

Scalability & Future Scope

In order to deal with real-time data from thousands of commercial drones, the system needs to mature from a basic simulator to a distributed, scalable architecture:

- **Real-Time Ingestion:** Utilize message queues such as Apache Kafka to process uninterrupted data streams from drones effectively.
- **Distributed Processing:** Divide airspace into regions and apply frameworks such as Apache Flink or Spark Streaming to detect conflicts in parallel.
- **Efficient Algorithms:** Eliminate brute-force checks and apply spatial indexing (e.g., R-trees) to decrease computation.
- **Secure & Compliant:** Provide encrypted data transfer and keep audit trails for compliance with regulations