Implementing Formation Control Strategies for Multi-Agent Autonomous Miniature Air Vehicles in Python Simulation

Dallin Cordon[1]

***Abstract*— A Python-based package built over the course of the Winter 2024 semester is expanded upon to include multi-agent capabilities and formation control strategies. Multiple miniature air vehicles (MAVs) are instantiated and assigned leader and follower roles. The lead aircraft is given specified waypoints while the follower seeks to maintain relative position and orientation with its respective leader. This pattern of leader and follower is augmented with an additional follower to formulate a Vic flying formation.**

I. INTRODUCTION

Formation flying is a technique where multiple aircraft fly close together in a coordinated manner. This approach allows aircraft to maintain specific positions relative to one another, forming predefined patterns or arrangements. Formation flying is commonly utilized in military operations, in aerial shows, and could be implemented in commercial aviation for fuel efficiency.

Over the course of the BYU Winter 2024 semester, students have developed a Python-based package designed to simulate small unmanned aircraft. Through this process, we have built in the necessary kinematic, dynamic, and control architectures to perform path planning, management, following, as well as lower level tasks like maintain trim state and implement autopilot schemes. The objective of this project is to expand the functionality of the simulation package by exploring and implementing formation flying capabilities within the simulated environment under the assumption of centralized control.

II. APPROACH

Our simulated aircraft package builds upon basic path primitives such as straight lines and circle trajectories. Vector fields are developed for each motion primitive to drive the aircraft into a stable, path-following trajectory which can be used by the path planner and path manager.

Taking advantage of these existing tools, we used simple math to define an offset distance and angle from the vehicle-1 frame roll axis to develop offset motion primitives that would keep the follower aircraft in proper formation with the leader.

While this path-offset approach worked well, the assumption of constant speed broke the follower's ability to maintain it's formation. To tackle this, we implemented two speed varying methods. The first was used during straightaways. The speed was proportionally altered from the leader aircraft's speed such that, as the aircraft strayed further from its desired offset location, the speed would similarly be altered.

[1]With the Robotics and Dynamics Lab in the Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA

Fig. 1: F-22A Raptors flying in Vic formation.

If the aircraft lagged behind, the speed would increase. If the aircraft was too far ahead, the speed would be reduced. We found this method to be simple and very effective to keep the aircraft in check and reduce any drift.

The second method whereby speed was altered was during orbital paths. In order to keep the formation together, the follower aircraft would either need to take a larger or shorter path based on the orbit direction and the relative lateral offset of the follower aircraft from the leader aircraft. This change in relative distance traveled meant that the leader aircraft would either quickly leave the follower behind or be left behind by the follower if speed remained constant. To tackle this, we used a simple equation keep the angular velocity of each aircraft identical as follows:

$$\omega_1 = \omega_2 \tag{1}$$

$$\frac{v_1}{r_1} = \frac{v_2}{r_2} \tag{2}$$

$$v_1 = \frac{v_2}{r_2} r_1 \tag{3}$$

A. Integration

To integrate my approach for formation control with the existing simulation package, we first created a launch file that could instantiate multiple aircraft from the MavDynamics class. Then, we created a path follower class based on the path follower from Chapter 10. This class was adapted to take in the leader's path and state, the desired offset distance and angle, and the follower's state. From there, the individual path primitives could be assigned and the commanded velocities adjusted.

The last necessary component for this to work was to create a viewer class that could take multiple instances of

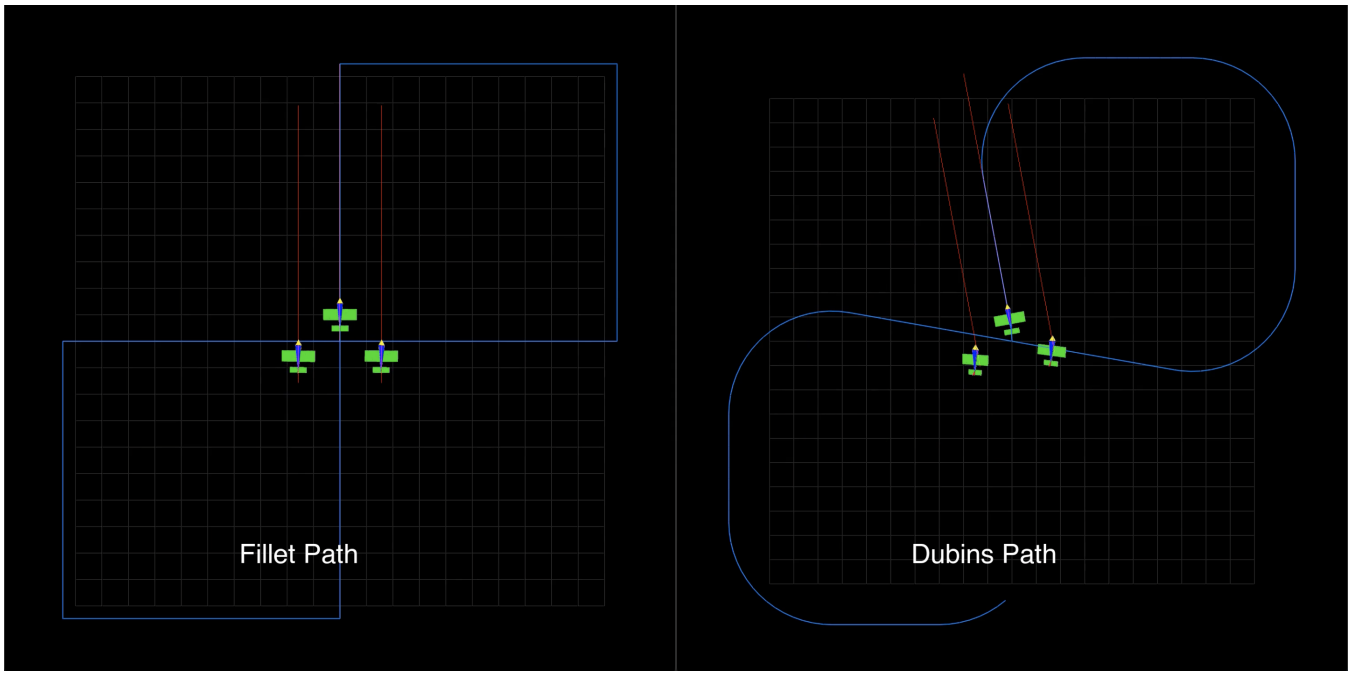

Fig. 2: Follower aircraft can maintain formation with the leader through both filleted and Dubins-based paths.

the MavDynamics class to show multiple aircraft at the same time as well as their paths.

III. RESULTS AND DISCUSSION

The formation control scheme worked fairly well. The follower aircraft were able to adjust their relative pose and orientation to match the formation assigned to them based on the leader. When the follower aircraft were started in initial positions that were incompatible with their goal formation positions, the aircraft were able to fly to their indicated locations relative to the leader. If the aircraft were too far ahead, they slowed down to allow the leader to catch up. If they were too far behind, they would speed up as expected to meet up with the leader.

A number of things could be done to improve this work. First, the leader's speed is set to the default we have used throughout the class; however, when making turns that force the follower into a tight turn, the follower has to slow down considerably, making it difficult for the follower to keep itself in the air. This was the largest difficulty we faced for this project. One potential solution for this includes adapting the leader's minimum orbit radius to account for the follower's minimum feasible radius without losing stability. Another might involve adapting the leader's speed so as to not drop below a minimum viable speed whereby we might have lift and stability for the follower aircraft.

The transition from an orbit-based path to a line-based path and back also presents an issue. The followers simply change from a line path to an orbit path and back when the leader changes its path. This often means that the follower aircraft are forced into a somewhat awkward phase transition. It would be better to have some method of adaptive smoothing between these transitions so the followers might finish out their orbits nicely.

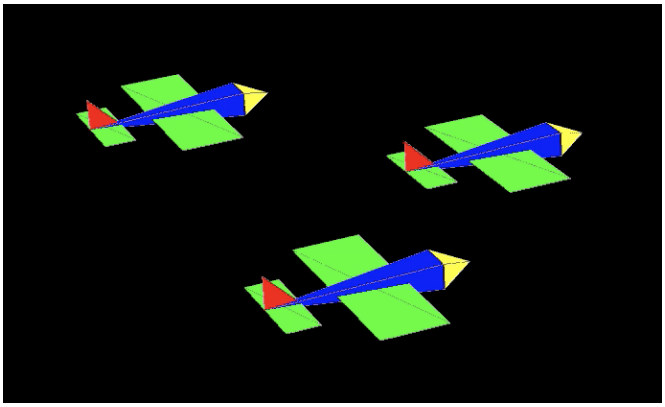

Fig. 3: Simulation aircraft flying in Vic formation.

IV. CONCLUSION

By assigning leader and follower roles to multiple MAVs, we have expanded the simulation package's functionality to include flying formations, wherein one lead aircraft guides one or more followers. The ability to set any aircraft as a leader to any other aircraft allows one to use multiple sets of leader-follower pairs to create finger-four formations or any other formation of interest. While our results show promising performance in maintaining formation integrity and relative positioning, there remain avenues for refinement and improvement. Addressing challenges such as speed synchronization during turns and smooth transition between different path types will be crucial for enhancing the realism and effectiveness of our simulation.