Project 3 Report: Quadrotor planning and control with VIO

Harsh Goel

I. INTRODUCTION

The goal of the project was to develop a complete planning and control system with state estimation using visual inertial odometry (VIO) for a quadrotor, enabling it to fly from an initial point to a goal point efficiently and without collision with obstacles or crashing. The planning and control systems for the quadrotor designed in Project 1.2 and Project 1.3 were without state estimation. Hence, with perfect information of the state of the system aggressive trajectories and controllers were feasible.

With state estimation, however, aggressive trajectories designed in prior projects are not feasible due to tracking errors in position and attitude of the quadrotor introduced by the Visual Inertial Odometry(VIO) that is sensitive to themotion of the quad-rotor. Thus, this report outlines the changes made in the trajectory generation and controls of the system, in order for the quadrotor to perform well under the constraints introduced by state estimation.

II. OVERVIEW OF CHANGES

Trajectory generation starts with an A* search with the Manhattan distance heuristic as in Project 1. Changes were made in timming the raw A* path, spline generation and controller tuning as compared to project 1.

A. Path Trimming

In project 1,intermediate points on a line segment defined by a direction vector are trimmed. In project 3, we use the Ramer Douglas Peucker(RDP) method to trim points. More specifically, RDP recursively trims points by avoiding points that are within a threshold distance between a start location and goal location. The start location and goal locations are recursively set to an intermediate point which is furthest away from the line segment. The points obtained from the RDP solution is further post processed to ensure that the minimum distance between any two points on the trajectory is within 2m. This is done in order for the spline generation in the following section to conform to the A-star path. Furthermore, this gives better control over deciding the time of flight for each spline.

Figure 1 show the difference in the trimmed trajectories. With this method, it is possible to trim more waypoints that increase trajectory planning time due to the optimisation as mentioned.

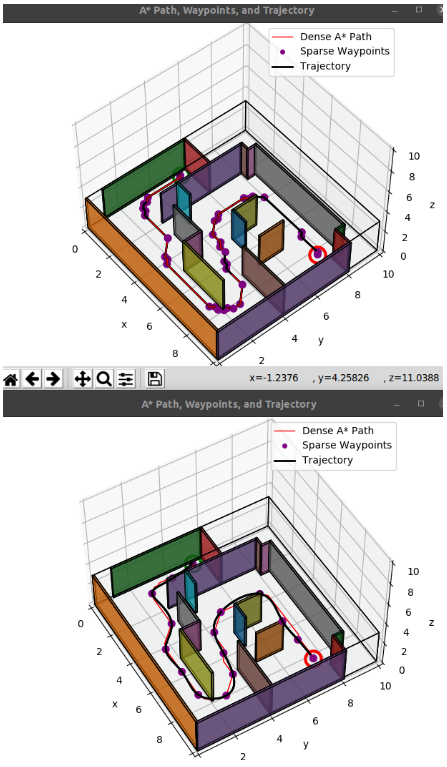

Fig. 1. Trajectory Generation Project 1 vs Project 3

B. Spline generation

In project 1, a naive constant velocity trajectory was used to plan a path for the quad-rotor. This wasn't optimal due to reasons discussed in section 3. As compared to project 1 we apply minimum-jerk trajectory generation to the post-processed A* path in order to compute a feasible trajectory for the quad-rotor.

For the flight, we also let the quadrotor hover for a small time of 0.2 seconds before executing the trajectory. This is done to ensure that the state estimation covariances settle before the quadrotor's flight.

The time of flight for each spline T_i is computed using the constant acceleration heuristic. Hence, given the Euclidean distance for 2 waypoints d_i, we have

$$T_i = \sqrt{\frac{d_i}{a}}, \quad i = 1, 2, \ldots, k \tag{1}$$

In our experiment, the acceleration $a = 4.5$. This has been furthermore modified to account for sharp turns. The time required for a spline is dilated by a factor of

1.2 if the angle between the current spline and the spline prior to it is greater than 60 degrees. This modification proved to be reasonable enough to pass the given test cases.

By formulating this minimum jerk trajectory with continuity ensues that the commanded acceleration and velocity profile of the quadrotor along all 3 dimensions is smooth. This is essential as the quadrotor's state prediction component of Kalman filter based state estimation relies on the accelerometer, and any discontinuities in the acceleration profile would lead to covariance drift.

Furthermore, such sharp changes in the velocity or acceleration of the quad also impact the VIO. As the VIO relies on feature tracking between time-steps to estimate the change in attitude and position, with sharp changes in acceleration some features would not be tracked in subsequent time-steps. This also causes the state estimation to break or drift. This is shown in 2 for the maze test case. With the naive solution, the quadrotor drifts due to the discontinuities in the commanded trajectory velocities.

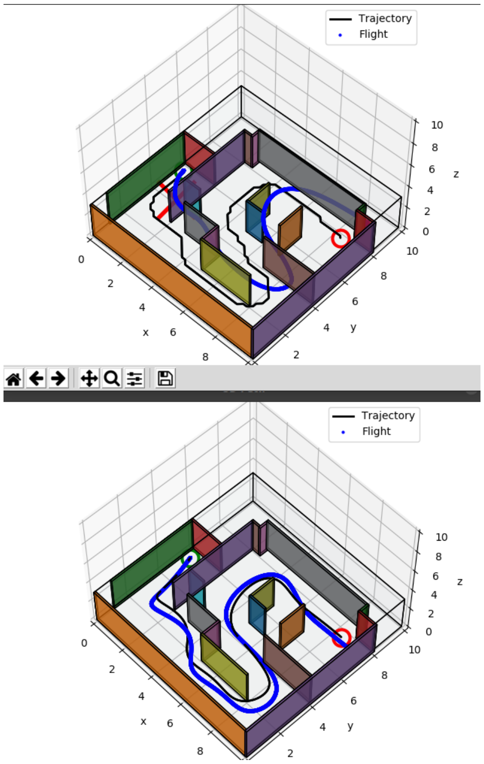

Fig. 2. Path taken by quadrotor Project 1 vs Project 3

C. Controller Changes

The Nonlinear Geometric Controller is used as described in Project 1.1. The controller had to re-tuned in lieu of the changes in trajectory generation and to ensure that the quad-rotor didn't make aggressive maneuvers while executing trajectories with high concavity due to which the VIO loses track of features.

Project 1.1 gains are as follows: Note that $K_d = 2\ \epsilon\ K_p^{0.5}$ where $epsilon$ is the damping ratio.

$$\begin{aligned} K_p &= [10, 10, 12] \\ \epsilon_p &= 0.85 \\ K_r &= [250, 250, 250] \\ \epsilon_\omega &= 0.85 \end{aligned}$$

Project 3 gains are as follows:

$$\begin{aligned} K_p &= [4, 4, 6.4] \\ \epsilon_p &= 0.70 \\ K_r &= [133, 133, 18] \\ \epsilon_\omega &= 0.40 \end{aligned}$$

III. RESULTS AND DISCUSSIONS

This section summarises the impact of the above changes as discussed above in the experiments conducted on window,maze and over-under maps given for testing. Majority of the changes have been made in the trajectory generation in this project as compared to project 1.3.

A. Robustness to Initial conditions

The above solution has been highly tuned and isn't robust to all initial error covariance conditions. Huge initial covariance errors in position may lead to different response trajectories due to the initial error in estimated positions. This could lead to collisions. By slowing the first spline component for the entire trajectory, this can be potentially compensated for but this solution was observed to not work for all test cases specifically for the stairwell test case on the autograder which the above implementation doesn't pass reliably.

B. Simulation to Real Gap

As observed from the controller gains in prior section, it is clear that the very high gains tuned for the simulation wouldn't work for a real quadrotor due to actuator constraints and intrinsic disturbances in the dynamics of the quadrotor. Furthermore, in a real quadrotor VIO that is dependent on tracked features from the camera will be influenced by lighting condition and the speed of the quadrotor (motion blur). This implies that the variances of estimates states from the VIO would not be constant.

In conclusion, the project solution attempts to address the complexities introduced by the state estimation and VIO. While, the solution fails to reliably pass the stairwell test on the autograder which would definitely require additional tuning in controllers as well as time of flight for the splines, other difficult test cases were reliably passed with varying initial conditions.