Individual Lab Report

Erik Sjoberg

Team C – Column Robotics

Rohan Thakker, Job Bedford, Cole Gulino

IRL 10

March 30, 2016

Individual Progress

Designed and installed light diffusion sheet and textured flooring

During flight testing in the NSH B-level hallway it became clear that our Iris+ optical flow system behaves poorly over the grey concrete surface and existing lighting. Building upon experience with our AR.Drone system, we knew that the sharp shadows caused by the overhead arc lighting can interfere with optical flow estimates, and I designed and installed a light diffusing material in order to reduce the impact of the shadows. The installed diffusor can be seen in the top half of Figure 1 below. Although this was extremely effective in removing any shadows from below the quadcopter, we still didn't have the performance we expected.

The second required component was highly-textured wooden "tiles" to increase the contrast of the grey concrete surface. After experimenting with a variety of configurations, the spaced pattern shown in the bottom half of Figure 1 below provided excellent performance as long as the quadcopter remains above the areas with wooden tiles. With the diffusor and tiles in place we were able to effectively test position controllers within our new enclosure.

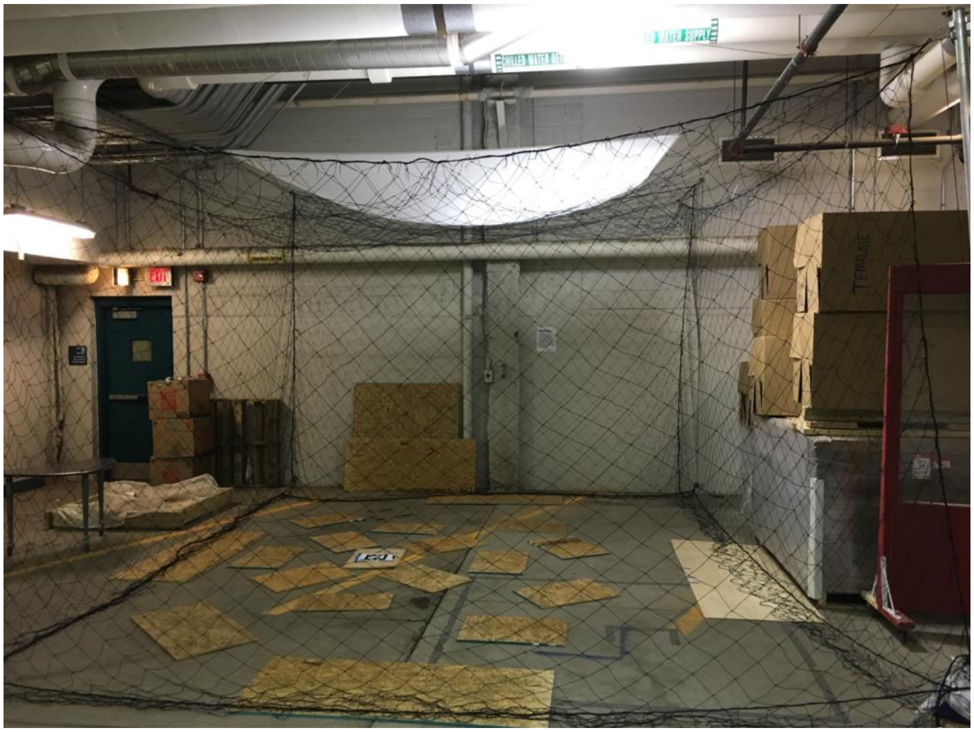

Figure 1: Light-diffusing sheet and textured flooring

Implemented and tuned PD velocity controller

Although our simple P-controller was successful at maintaining a relatively stable position, settling times were too long and increasing the gain resulted in oscillations. I implemented a PD controller to enable higher gains and damp the resulting oscillations. Figure 2 below shows the successful position and command velocity results of a square flight pattern with the new PD controller. The flight successfully followed the offset pattern (0,0) -> (+0.3, 0) -> (+0.3, +0.3) -> (0, +0.3) -> (0, 0) starting from the location (-0.28, 0.25) using a P gain of 4 and a D gain of 1. Command velocities peaked to +/- 3 m/s, but position is still able to quickly reach the target setpoint with minimal oscillation.

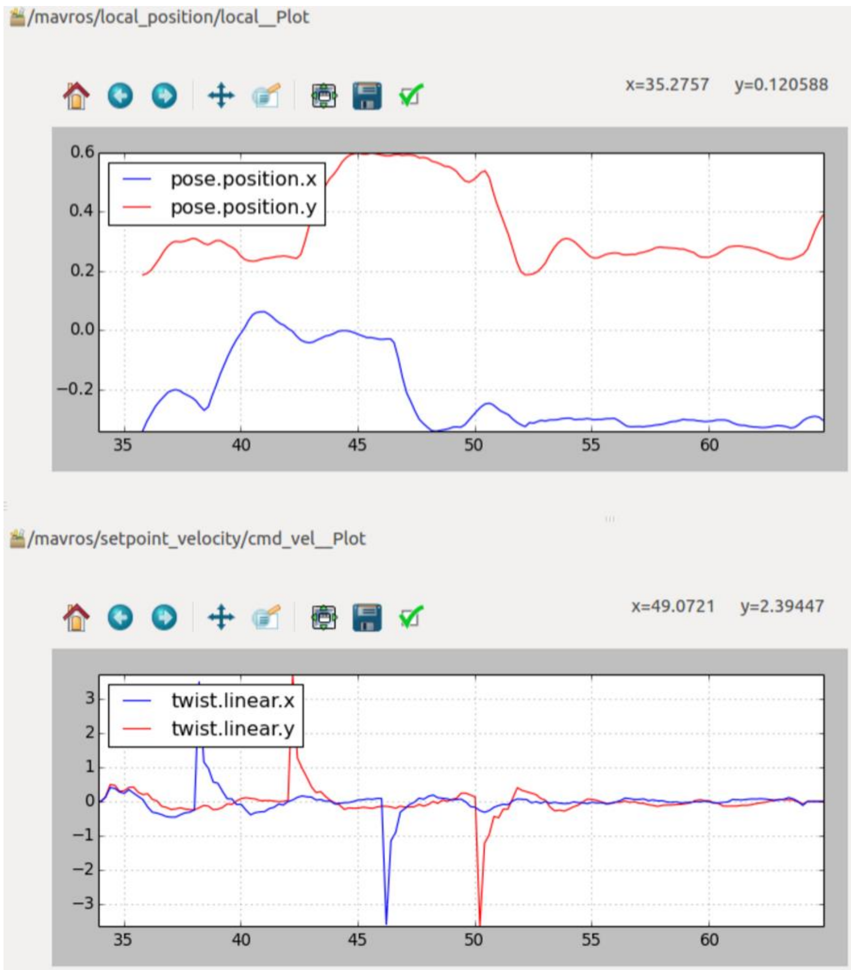

Figure 2: Square flight pattern under enhanced PD Controller

Position holding also exhibited very good stability with the new PD controller. As can be seen in Figure 3 below, we were able to maintain a very steady position of +/- 10cm over approximately 4 minutes

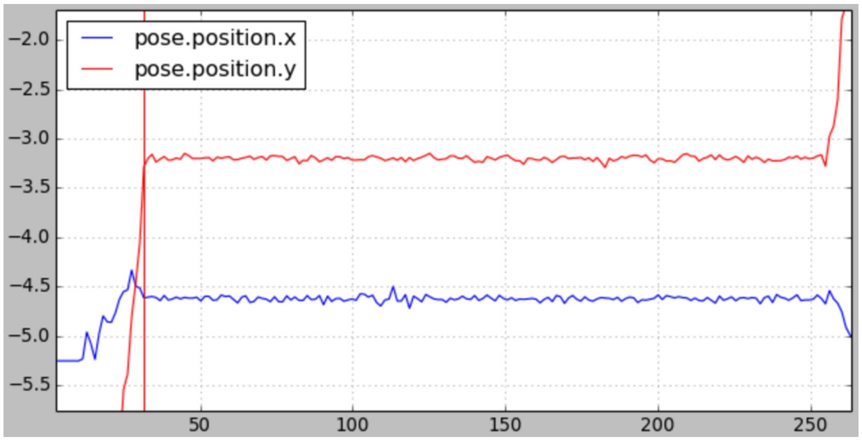

Figure 3: High-accuracy (+/- ~10cm) position hold over approximately 4 minutes

This high positional accuracy has enabled arbitrary search patterns similar to those shown in our fall validation experiment and gives us a good base from which to build out further functionality.

Implemented a parameter-based tactical planning system for scripted behaviors

With our base controller working properly, I then moved up the stack to implement the infrastructure which will power our tactical and global behavior. I leveraged the ROS Parameter server to serve as a “blackboard” of shared state variables such as controller gains, setpoints, and event flags which enable us to easily script behaviors for the entire system from nodes written in Python instead of relying entirely on hard-coded C++ behaviors. Using these setpoint parameters, I was able to script various movement patterns and conditional behaviors, including manual control of the sequence start time from the hand-held controller as well as automatic landing after completing a search sequence. This system is extremely extensible, and will serve as the foundation of the rest of our system’s logic and behavior code.

Challenges

Optical flow sensitivity

The optical flow system on our Iris+ quadcopter was developed for outdoor environments, and as a result it has an unfortunately small field of view resulting in poor performance during low-altitude flight over textured surfaces. This required a variety of workarounds and testing to enable stable flight in the NSH B-level hallway which were in the end successful. Although off-the-shelf components do result in easier system integration, this has been an example of how the lack of control over the design specifications can result in problems.

Codebase integration

With multiple teammates now working on the same code base, we are now facing challenges in integrating various branches of our git repositories which work independently, but drift apart during development. This was the problem which prevented this week's demo from performing as well as it could have. More discipline in how we use feature branches should reduce the likelihood of this being a problem in the future.

Teamwork

Teamwork went especially smoothly this week. Cole and Rohan completed theoretical work on our Kalman filter setup, and Job joined them in finally completing our closed-loop control of hovering over the april tag. This split of work was effective as it allowed me to focus on the basic control stability and behavior scripting framework. Some late-night work meant that I was unable to integrate all of their work successfully this week, but an enormous amount of progress was made on the individual pieces.

Plans for Upcoming Work

Over the next two weeks I will be working to integrate our completed components into a complete spring validation experiment which we can then begin to enhance. Once the base spring validation experiment is working reliably, I will switch to focusing on our higher level goals of global pose estimation via RGBD SLAM as well as obstacle avoidance.

Meanwhile, Rohan, Job, and Cole will be focused on implementing yaw control and then increasing the precision of our landing sequence.