Progress Review 12

Cole Gulino

Team C / Column Robotics

Teammates: Job Bedford, Erik Sjoberg, Rohan Thakker

ILR # 11

April 12, 2016

Individual Progress

During this PR, the main individual work that i did was on the integration of the software systems. Erik had been working on the script for the integration in python. This allows us to prototype rapidly in python without the need to compile. My previous work was in the individual features in C++.

This sprint, I focused on getting the small pieces integrated in python. My main individual contribution was to work on the subscriber in python for the april tag information.

In the las PR, I worked on getting filtered estimates of the April Tag, and this PR I worked on the logic for state shifting.

Figure 1 shows an example of the state machine logic implemented in the code:

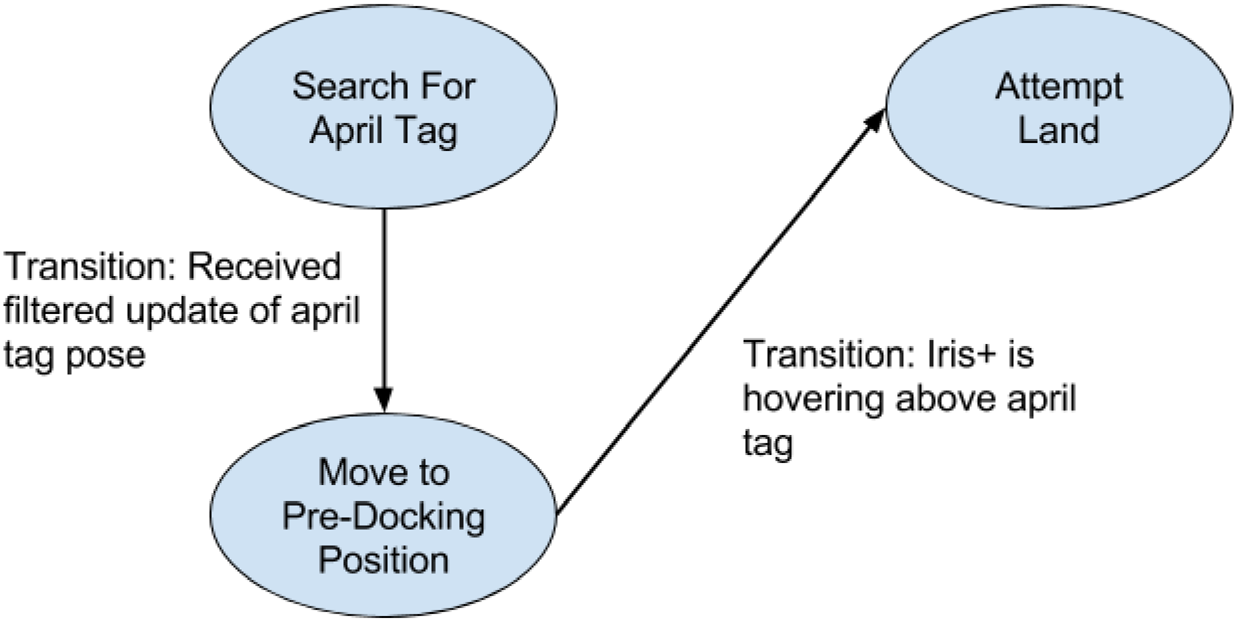
Search For April Tag
Attempt Land
Transition: Received filtered update of april tag pose
Transition: Iris+ is hovering above april tag
Move to Pre-Docking Position

Figure 1) State Machine Logic

My individual contribution this PR was on the transition from Search for April Tag -> Move to Pre-Docking Position. In our code, we use the ROS parameter server to set specific parameters. I wrote a subscriber, [1], that writes a parameter whenever it gets an update of the filtered april tag estimates. This parameter provides the transition between April Tag -> Move to Pre-Docking Position states.

I also wrote the logic in the main python script, [2], in order to ensure that the drone switches states

Challenges

The major challenges in this PR were all based in the integration of the various subsystems. Because we were shifting to the python script from a serious of C++ nodes, we needed to shift over our logic.

The shift in logic was the biggest challenge that we faced during this PR. In order to ensure that the system was working completely, we needed to run multiple tests. Figure 2 shows a simple test of holding in place which tested our hover over april tag subsystem.

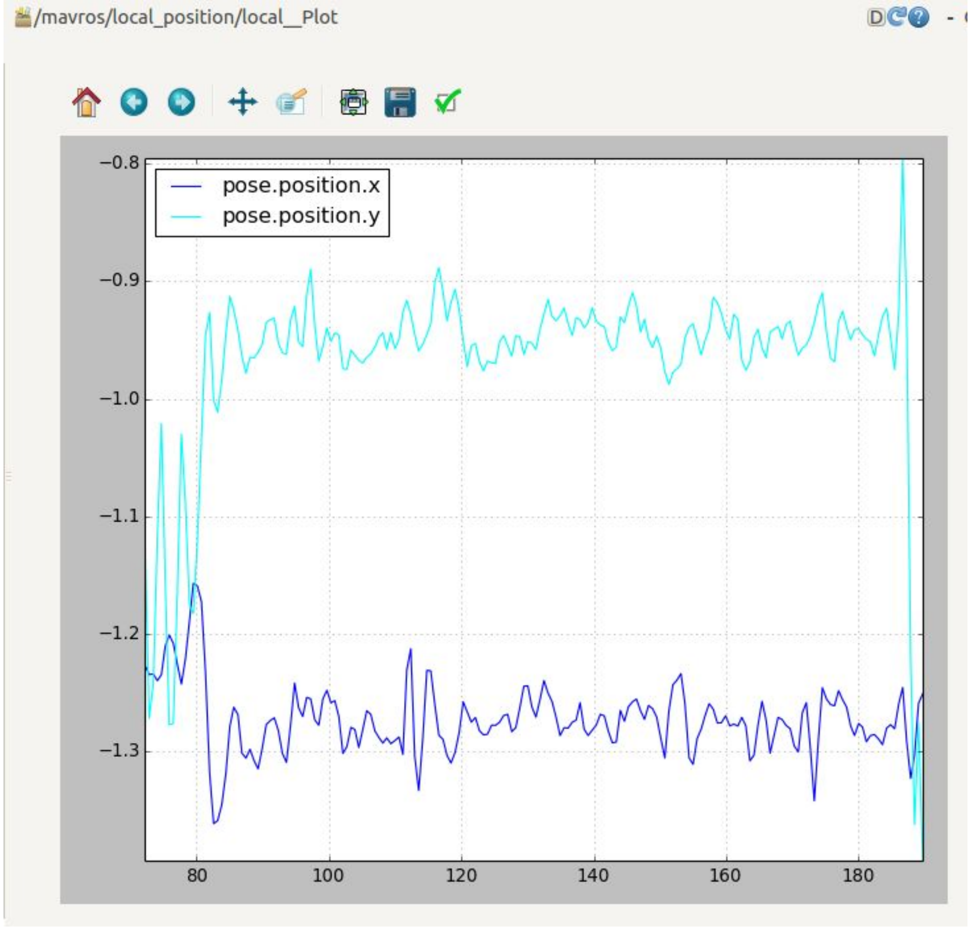

Figure 2) Position Estimates While Hovering Over April Tag

In Figure 2, the independent axis is time in seconds and the dependent axis is distance in meters.

As can be seen in Figure 2, we were able to get consistent hovering over the april tag whenever we were hovering. This shows that our PD controller works accurately whenever we have accurate updates of our position either from good optical flow readings or from the april tag.
This is why the state transitions needed to be so reliable. Their reliability was another major challenge that we faced in this PR. In order to accomplish this, we placed many edge case checks to ensure that the setpoints and position estimates were accurate. For example, we ensured that the filtered april tag updates were not too far apart, which would make their average unreliable.

Even though we had our subsystems working, it took considerable effort to complete the integration

Teamwork

As usual, we had plenty of teamwork throughout this PR. Me, Rohan, and Job worked extensively on the integration. Erik worked on the system that glued together the integration. He got the ROS parameter server working and the python script skeletons set up. He also perfected the demo and worked through the bugs to ensure robustness of the demo.

Me, Job, and Rohan were tasked with using Erik's software architecture in order to integrate all of the subsystems that we had completed in the previous PR.

Figure 3 shows the output from another of the subsystems: search

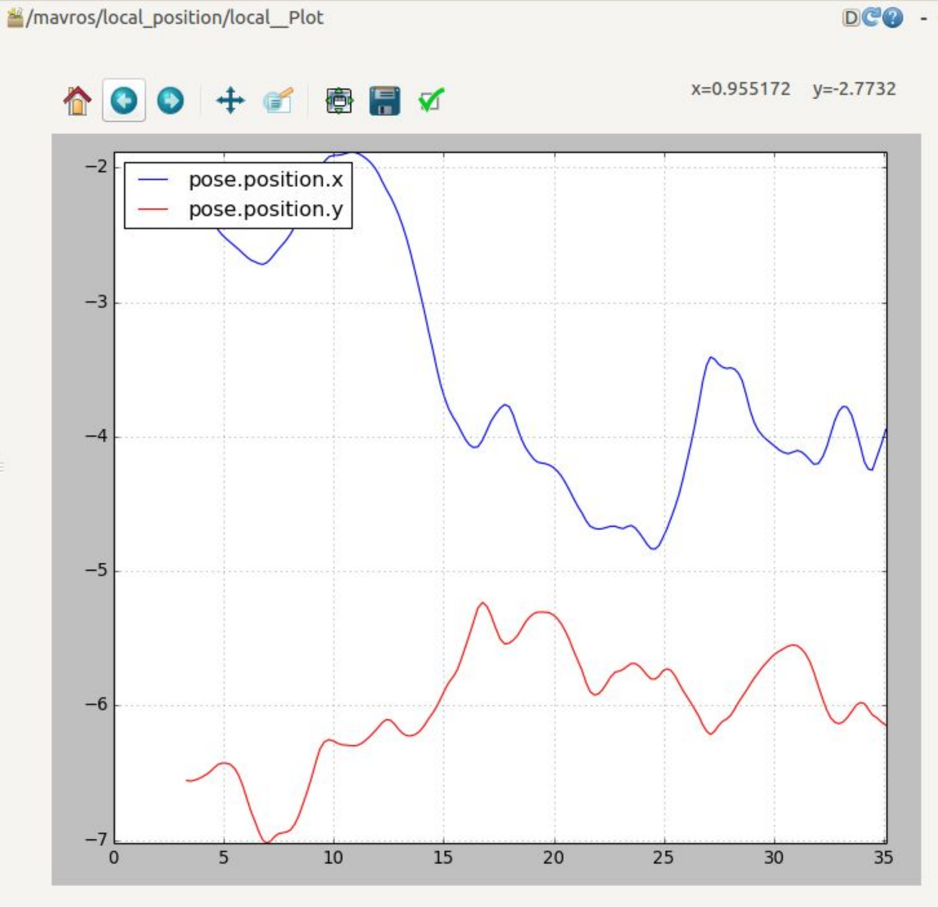

Figure 3) Cone Search Subsystem

In Figure 3, the independent axis is time in seconds and the dependent axis is distance in meters.

Figure 3 is a depiction of one pass of the cone search. It is an example of the work we did to test each individual subsystem similar to what we did in Figure 2.

Once we had tested each subsystem, we then moved to the python script [2] in order to integrate the subsystems.

The result of our work can be seen in this video [3].

Future Work

We feel very confident in our success criterion for the minimum scope of our project. There are a few things that we would like to accomplish for the SVE and SVE encore.

Erik will be working on getting the SLAM system working onboard with the quadcopter. He will have to ensure that the SLAM packages we have looked at compile and run on our system. Once he has done that, we will use a ROS wrapper to provide the updates for us.

Rohan and I will work on providing the position updates from SLAM to the EKF of the drone.

Job will continue working on the landing to improve its robustness and precision.

Resources

[1] tag_check.py

[2] search_hover_land.py

[3] Search and Land Iteration 1