

**Progress Review 9**  
**Project Pegasus**

Tushar Agrawal

Team A – Avengers Ultron  
Teammates: Pratik Chatrath, Sean Bryan

**ILR #8**  
February 24, 2015

## 1. Individual Progress

After the last Progress Review, the UAV was capable of being guided by the onboard Odroid.

### **Hover over moving marker**

Basic position guided autonomous motion was demonstrated in the last PR. A basic test for hovering over a static marker was also shown. The test assumed the marker does not move. The marker was only seen for a few seconds to identify its position, which was sent to the UAV to hover.

Further, I adapted the code to dynamically track the marker and hover over it as it moves. The first test exposed a small problem. The UAV would come close to the markers position but slowly drift away until it could not see the marker again. (as seen in the video <https://www.youtube.com/watch?v=vm3XfKOhDLA>). After further analysis, I realized I had considered the wrong Z-direction for the global frame (from point of take-off). This led to wrong y offset values for the position of the marker and hence, the UAV was basically being repelled in the Y-direction in the previous video.

After making the appropriate corrections, the UAV was able to approach and maintain its position over the marker. But, as the exact position of the marker was being forwarded to the UAV, it would tend to turn frequently and erratically to compensate for minor variations. I added a small tolerance into the values sent to the UAV. A change was forwarded to the UAV only if the change in X or Y was atleast 0.5m. As seen in the demo video (<https://youtu.be/h8XrTq4-k1E>), the UAV was able to hover over the marker and follow it as it was moved in a square.

### **Test navigation stack control medium on a simulated UAV**

One of our major risks in the current system was the integration of the navigation stack with the UAV flight control. After completed by initial task, I decided to move forward on this front. This task was two-fold. Firstly, the odometry data from the pixhawk had to be forwarded to the navigation stack for dynamically calculating paths to the goal. Secondly, the output velocities need to be sent to the UAV and make sure the UAV responds as expected.

I started working on the second subtask. To confirm that the command velocity format as outputted from the navigation stack can be used by the UAV, I created a test ROS node which would send preset linear and angular velocities to the simulated UAV. When the UAV was in guided mode, I observed that the linear command velocities worked as expected to move the UAV in the correct direction with the given velocity, but, unexpectedly, the angular velocity about z (yaw velocity) had no effect on the UAV. After reading much documentation online, no apparent reason could be seen. A snapshot of the simulation (Image 1) shows the UAV being commanded to move with 1m/s velocity in negative X and positive Y axis, and 1 rad/s angular velocity in Z, which the UAV does not respond to.

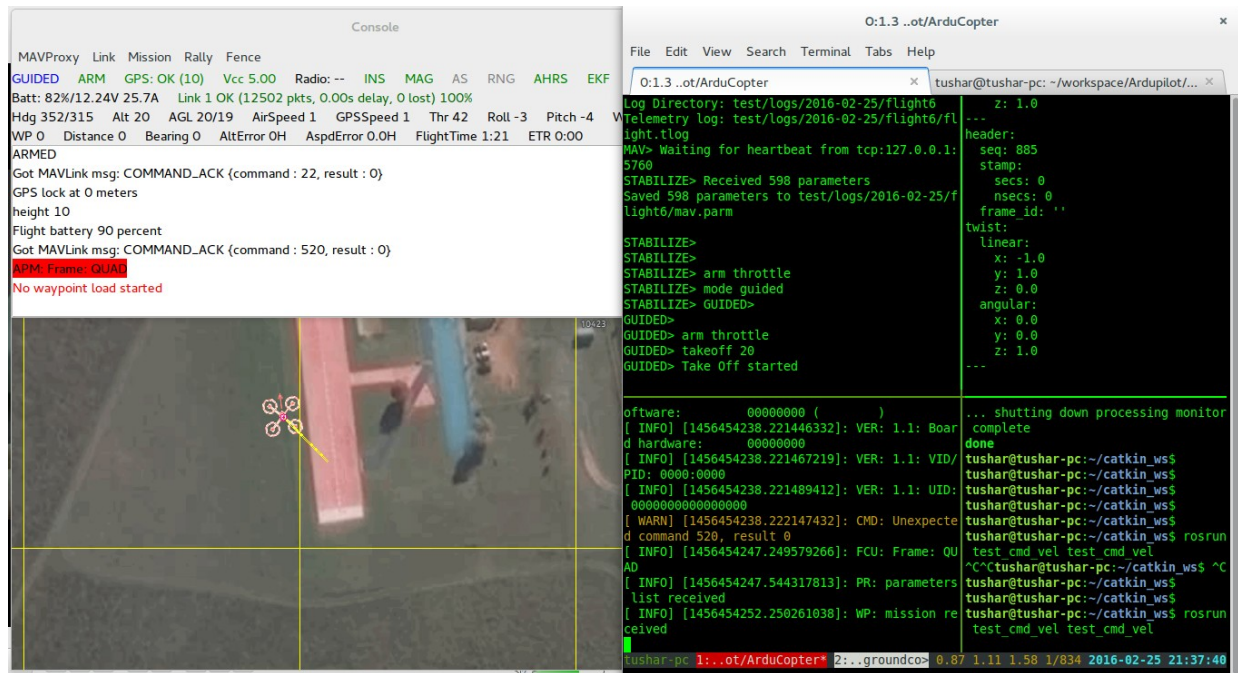


Image 1: Test for the response of the UAV to command velocity inputs

Finally, considering this anomaly might be limited to the simulation, I tested the same thing on the UAV with the exact same results. We expect to find a solution or at least a work around for this issue by the next PR.

## 2. Challenges

The first challenge I faced when implementing the “hover over moving marker” functionality, was the issue of correct frame definitions. At first, I converted the camera frame data (position of apriltag) into the UAV frame, from where it was converted into the ground frame using the position of the UAV in the ground frame. At first it was assumed the ground axes were oriented the same way as the UAV, which turned out to be wrong when I manually flew the UAV and tested for changes in the UAV position.

The second issue was the constant movement of the UAV to correct for minor variations, which was easily fixed by adding a tolerance in the goal position.

Finally, the current challenge is to figure out why the UAV is responding to the linear velocities sent to it but not the angular (yaw) velocity.

## 3. Teamwork

Due to relatively fewer days with clear weather conditions, we were only able to test the UAV outside a few times. We tested and finalized the marker following algorithms though those. Owing to the high risk of integration issue with the navigation stack, I decided to try integrating the Navigation Stack output format with the UAV. And Pratik would look at integrating the odometry data from the UAV to the Navigation Stack.

I worked with Pratik to help him relay one rostopic into another and a basic node to convert the command velocity data into a format acceptable by the UAV.

I also worked with Sean to control the NicaDrone through the pixhawk on the UAV. Due to lesser number of channels available on the Radio Controller, the NicaDrone control was demonstrated using the Mission Planner UI, which should ideally not be a problem as the NicaDrone would eventually be controlled through the Odroid.

#### **4. Future Work**

The current development progress is satisfactory based on our initial plans. For this PR, we will try to mitigate our highest risk item, i.e. the integration of the UAV with the navigation stack. Me and Pratik will continue to work on the current issues pertaining to this task. We will also set up the Odroid with the relevant ros packages for running the entire control stack.

In addition, I would also work on autonomous takeoff and landing for the UAV. This when combined with the current system, would be able to land on the marker to drop packages.

Also, I will work with Sean to set up the control of the NicaDrone using the Odroid.

As a result, we will be able to give a demonstration of autonomous A to B motion using the navigation stack in addition to autonomous take-off and landing through the Odroid.