

**Progress Review 8**  
**Project Pegasus**

Tushar Agrawal

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

**ILR #7**  
February 11, 2015

# 1. Individual Progress

After the last Progress Review, we had the UAV connected to the Odroid, which had mavros running for observing the state of the UAV and sending basic commands to it. After the first few tests for autonomous A to B Navigation using the Odroid failed, and weather continued to deteriorate, I decided to test the system on simulation to validate the software.

## Setup SITL for simulation

The recommended system for simulating the UAV with Ardupilot framework is the SITL (Software in the Loop) simulation environment. I set up the basic SITL framework on the laptop. This gave me the basic MAVLink connection to the simulated UAV. MAVlink commands could be used to control the UAV in simulation. As I wanted to use the ROS framework (Mavros) to control the UAV, I setup mavros to connect to the simulated UAV. Following that, I could see the simulated UAV state through mavros.

I tested the exact scripts as I was trying on the real UAV and realized the simulated UAV also rejected the mode switch for autonomous navigation. Hence, I realized it was a software or an interface protocol issue rather than a real world issue (like GPS fix or hardware problem). On further testing, I realized that the takeoff command needs to be sent separately to the UAV to takeoff autonomously and once it is in the air, it can receive autonomous waypoints for motion. This resolved the A to B navigation issue on the real UAV as well.

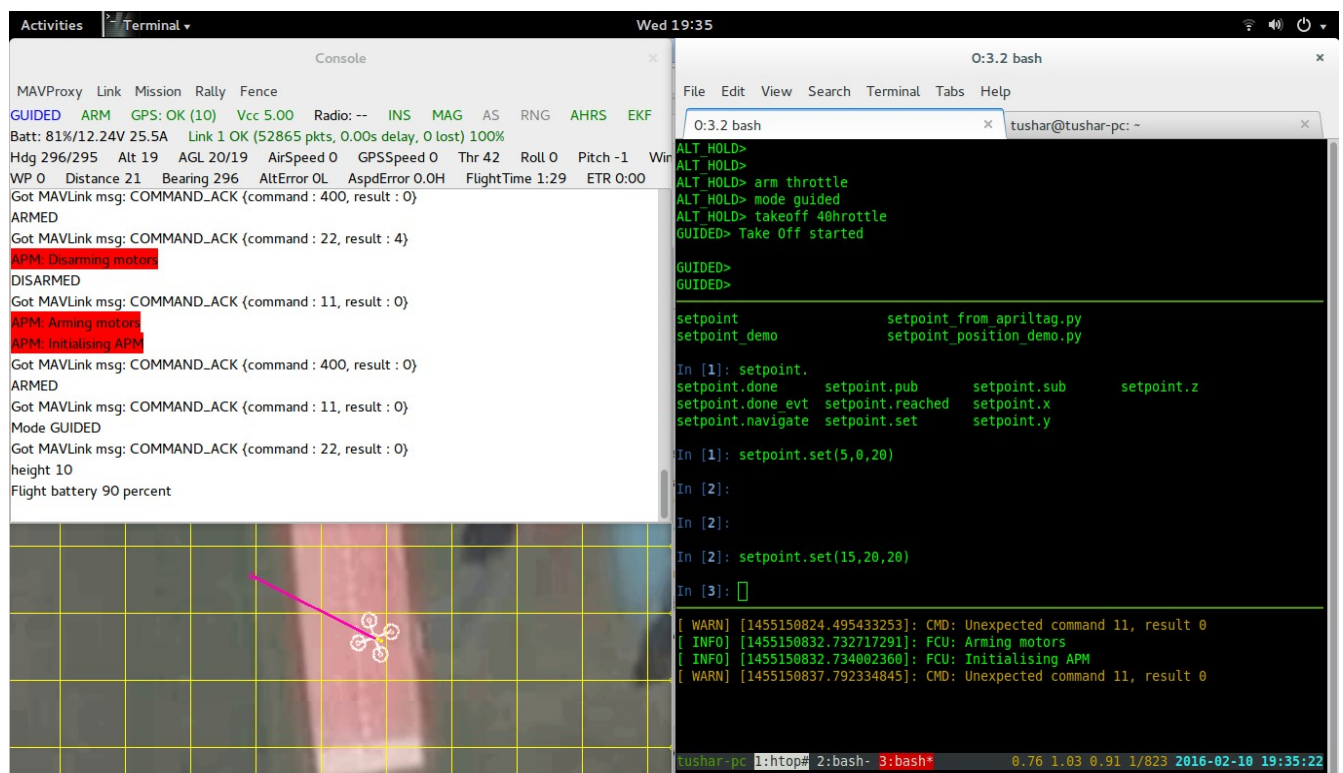


Image 1: Software in the loop (SITL) simulation test for UAV

## Autonomous Navigation from A to B

As mentioned, after mavros setup, I tested some basic programs to continuously control the position of the UAV with respect to its origin position. This failed multiple times until I tried the same thing on simulation and realized that takeoff had to be done separately (simple command on mavlink but requires exact GPS position on mavros which is not ideal). Currently takeoff and landing are done manually using radio controller.

Finally the autonomous navigation worked as expected barring a few minor issues as explained in the challenges section. The video (<https://www.youtube.com/watch?v=ps0AnLxfEpo>) demonstrates this. The UAV takes off using manual RC control, then, the mode is switched for autonomous navigation. The UAV stays at the original point for 2 seconds and then moves ahead (UAV X axis) by 4m, and increases its altitude by 1m. It then holds its position until mode is switched back and the UAV is manually landed.

After landing, the origin frame of the UAV is sometimes displaced and hence when the same test was done repeatedly, the UAV flew 4m towards a slightly different direction (approximate error of 20 degrees). Although hovering over marker would be a different scenario as the marker position is relative to UAV and is converted to inertial frame (with the takeoff point as the origin)

## Integrate navigation with marker detection – Hover over marker

There was considerable uncertainty in positioning due to uncertainty in origin frame after multiple landings. But to hover over the marker, the position of the marker obtained is w.r.t the UAV, hence the uncertainty might not affect that. I modified the code for Apriltag detection to fork another thread for sending continuous position updates based on apriltag location. It used a shared memory space for transferring data from apriltag detection thread to ros thread. It also read the current position of the UAV and used that to convert the coordinates of marker wrt UAV to global frame.

Due to multiple mutex locks at multiple positions, the program performed became very slow at apriltag detection. Due to that, and other factors, I split the functionality into multiple nodes. Apriltag detection just published the local apriltag position and another node setpoint\_from\_apriltag converted that to a global setpoint to which the UAV was sent. (refer image 2 below)

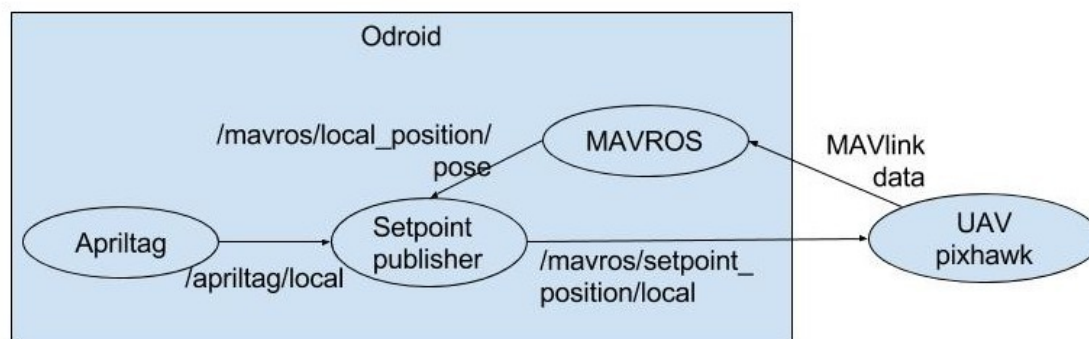


Image 2: ROS Node architecture for autonomous hover over marker

The results were very promising. As seen in the video (<https://www.youtube.com/watch?v=x94lj6qUGSQ>), after the UAV is put into autonomous mode (audible beep), the UAV positions itself over the marker and maintains position at 18m height until its taken back to manual mode and landed.

## **2. Challenges**

The first challenge I faced when trying autonomous navigation using the guided mode was that the UAV would not take off based on the positions sent using the program. The UAV did not take off, nor did it try to maintain position. After trying this in simulation, I realized the takeoff had to be done using a separate command or manually. For this PR, I concentrated on accurate position estimates instead of takeoff, hence, I went with manual takeoff for further testing.

Another issue was seen when A to B navigation was tested multiple times. The XY axes rotated considerably around the Z axis after each landing. This led to different positions for every time the same test was performed. But this issue was resolved when the position estimate was obtained w.r.t the UAV frame and converted into global and sent to the UAV.

Another issue which is yet to be resolved was appropriate realtime access or logging for debugging. A wireless hotspot created from my cellphone was used for wireless access into the Odroid to see the status. But this had intermittent issues. During some tests, the wireless worked perfectly but crashed entirely during others, due to which, it was difficult to keep track of how the system was performing. I plan to work on this aspect by the next PR

## **3. Teamwork**

Due to good weather conditions, we prioritized testing outside over making mechanical modifications to the UAV. Hence, Sean was still waiting to finalize all mounting for the UAV, which shall be happening in this PR.

Pratik has gotten the navigation stack setup to the extent that we can understand the requirements for integrating it into the autonomous navigation subsystem.

Finally, additional backup parts were ordered and received, as we can finally confirm the x8+ platform to be appropriate for our application. This included spare landing legs and GPS module with mast.

## **4. Future Work**

Based on our current plan, for the next 2 weeks, I was supposed to work on hovering over marker. Due to better weather conditions, this was accomplished during this PR. I shall work on making this system more robust for this PR. I will setup a logging mechanism for quick debugging. If weather permits, I shall also test if the UAV follows a moving marker.

In addition, Pratik and I will also work on setting up the environment for integrating the navigation stack into the autonomous system.

As the UAV shall be available, Sean would aim to finish up most of the important mounting tasks and with my help, setup control of Nicadrone through the Radio Controller.