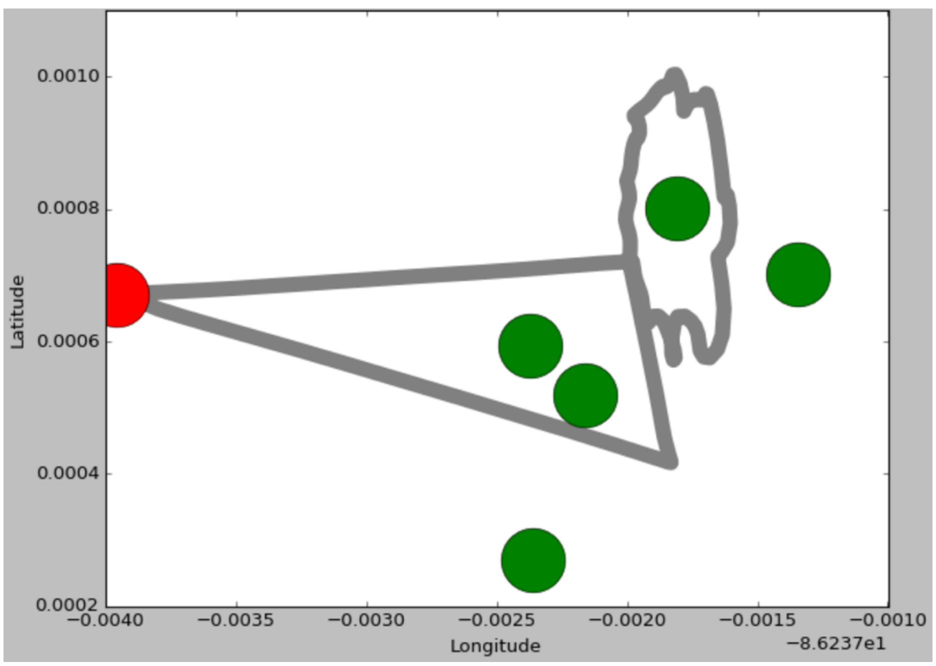
Justin Garrard

HW2 Report

For this experiment I included 6 coordinates to represent the 4 tree obstacles and starting position described in the project document. These coordinates are reported in Appendix 1. Note that Tree 2 has two coordinates because it had the equivalent foliage of two trees. Each flight in the experiment was run with the drone flying at 10m off the ground.

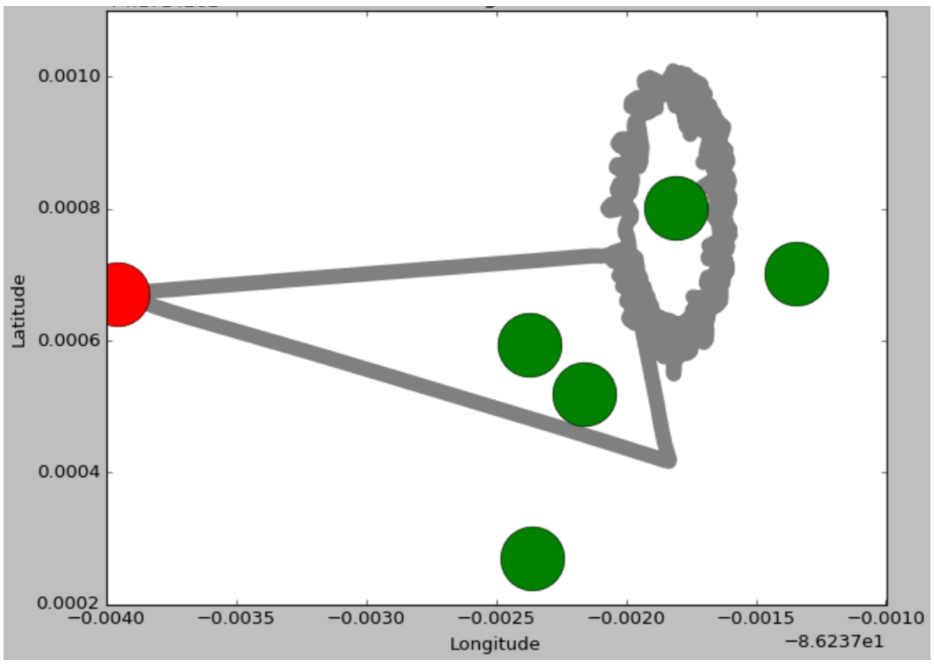
In every flight, the drone uses waypoints to arrive just southwest of Tree 3, then uses a variety of NED vector approaches to fly the circle. Starting from an angle of 250 degrees counterclockwise from East (the position upon arrival after using waypoints) the drone calculates points on a theoretical circle around the tree's coordinates and issues NED vectors to reach those points. Flight 1 locates a target location around the circle every 10 degrees, calculates a NED vector to that location, and flies to it for a duration of 1. The result is depicted in Figure 2 below, and it took 181.690 seconds to complete the flight. The path around Tree 3 takes the shape of an ellipse, and the NED vectors clearly stray from their target locations and require dramatic readjustments.

Figure 1 – NED (step=10, duration=1)



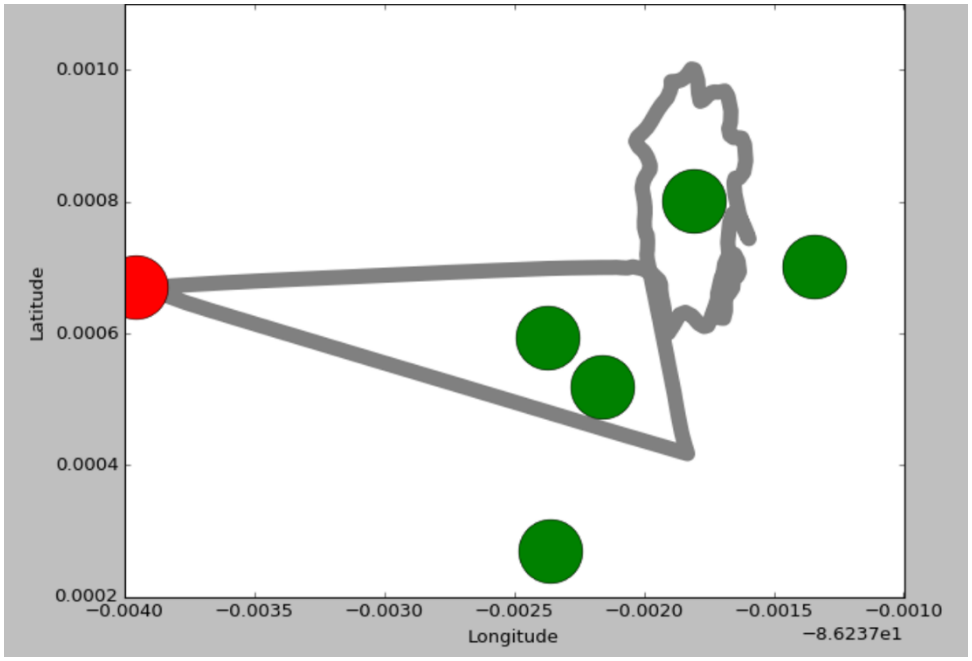
Flight 3 steps around the circle 1 degree at a time, using NED vectors to fly toward each of the 360 theoretical positions for a duration of 1. This took 703.992 seconds and the resultant path is shown in Figure 3. The NED readjustments around Tree 3 are denser which causes this path to be less smooth. The increase in the number of target locations generated causes the drone to swoop back and forth to get in target range of the location much more often which creates the sporadic path and causes the flight to take so long.

Figure 2 – NED (step=1, duration=1)




Flight 4 steps around the circle 10 degrees at a time, but flies toward each point using a NED vector with a duration of 4. This flight took 206.102 seconds and can be seen plotted in Figure 3 below. The increased flight time and slightly more sporadic path from Flight 1 are caused by the drone flying to long for a given NED which puts it out of position and causes more readjustment NEDs to be issued.

Figure 3 – NED (step=10, duration=4)




The parameters of the first flight were best when weighing both speed and smoothness. Calculating target locations every so often takes a lot of burden off of wasting time issues readjustment NED vectors. In my experiments I found that the duration value for the vector could not be a floating point value. I believe my results would have been better if I could have issued update NEDs more often, but I was bounded by 1. My theory is that I was flying too tight a circle, the radius around the tree was too small, for small incremental updates to target positions to be precise. This would also explain the ellipse pattern of the flights. The drone was coming in with existing velocity and the turn radius around Tree 3 was too tight to make quick turns. By the time the drone began to slow down, it was already turning back to head south. Given more time I would have liked to generate a function that flies an ideal circle of any radius using a related rate between the degree and the values of the North and East vectors. It should theoretically be possible to do without hitting target locations, but my research to this end had to be set aside to complete the experimental requirements.

**Appendix 1 – Obstacle Coordinates**

| Obstacle | Coordinates |
|---|---|
| Launch Position | 41.714870, -86.240956 |
| Tree 2 | 41.714720, -86.239160 |
| | 41.714794, -86.239374 |
| Tree 3 | 41.715002, -86.238809 |
| Tree 4 | 41.714470, -86.239362 |
| Tree 5 | 41.714902, -86.238346 |

**UTM Summary**

The UTM is NASA's approach to establishing a framework to begin safely testing the deployment of dense and beyond line of sight UAS research and commercial efforts. It is unsafe for individuals to develop drone applications that might fly over populated areas without considering how other drone traffic in the area might affect their own UAS. However, how does someone go about testing that kind of software? NASA proposes stratifying geographic regions into 4 classifications based on their population density and geographic features. TCL 1 is rural, flat land with few building or people around, while TCL 4 is used to classify bustling metropolises. TCL 2 and 3 fall in between. It would be unsafe to deploy drones with software which takes them out of an operators line of sight to TCL 4 because it could collide with tall buildings and/or cause harm to pedestrians. Therefore, NASA prescribes deploying tests in TCL 1 or 2 to test for proper drone behavior with risk of damage or injury.

This UTM framework was used to test the operation of many drones, in and out of an operator's line of sight, in a single area, and came back with 4 key findings. First, operators of drones really have no way of knowing where other operators in their area are or where they are headed. To encourage safety there needs to be a universal "flight plan" that operators can watch. Second, different hardware reported altitude differently. This could cause drones to collide when the operators thought they were avoiding each other. The equipment should be standardized. Third and fourth, information about weather and operational plans were not always consistent across weather and GIS and UAS systems. Operating under perceived different conditions in the same area causes unpredictable and dangerous behavior.