Estimation

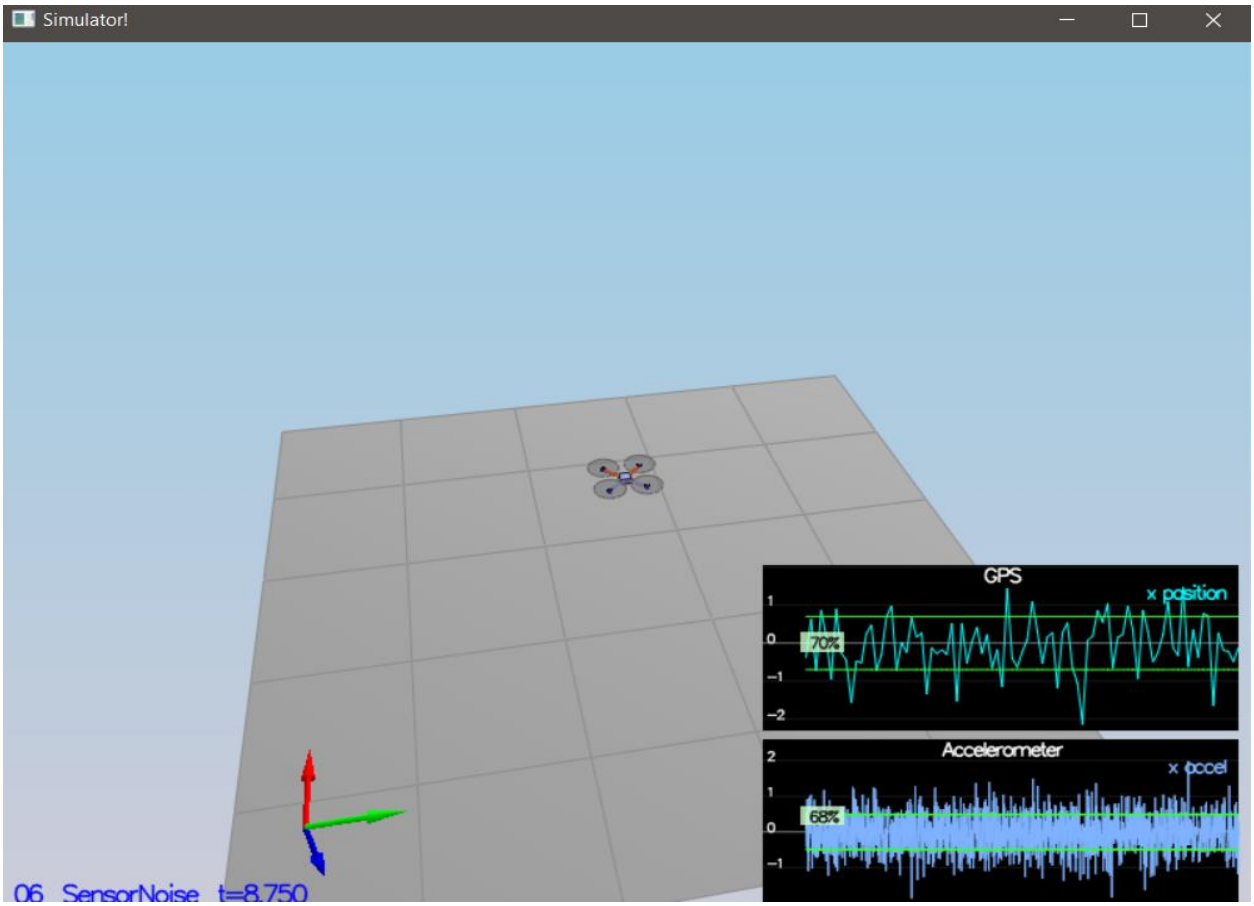
This project is divided into different tasks. Each task must be completed successfully in order to pass this project. The different tasks are mentioned below:

1) Sensor Noise
2) Attitude Estimation
3) Prediction Step
4) Magnetometer Update
5) GPS Update

Sensor Noise:

This step adds realism to the project as we now account for the noise in sensor readings. First we had to run noisy sensor scenario to record some sensor data. GPS X position and accelerometer X measurements were recorded and analyzed to calculate the STD.

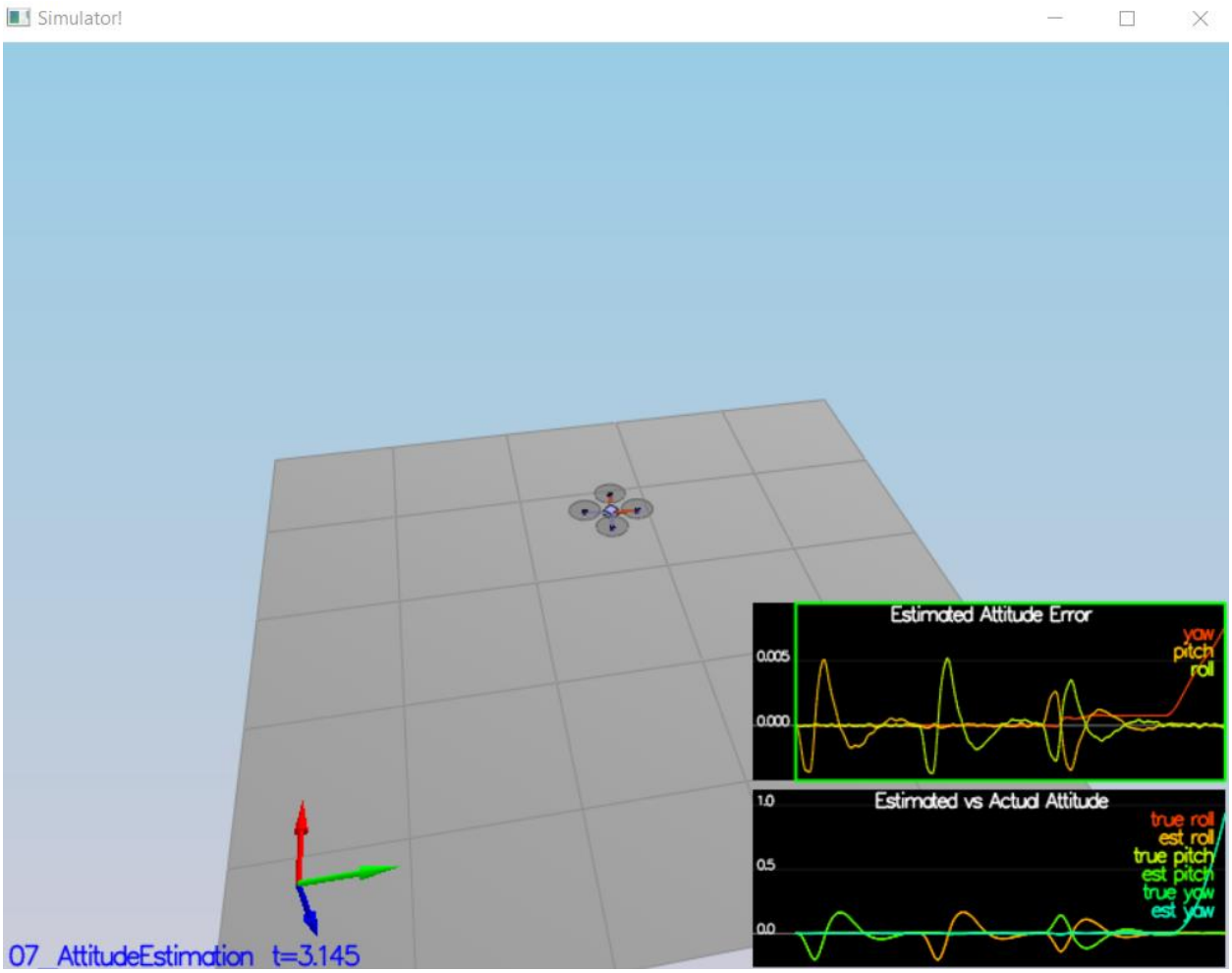
The success criteria required the standard deviations to accurately capture the value of approx. 68% of the respective measurement. Below is the figure of this scenario:



Attitude Control:

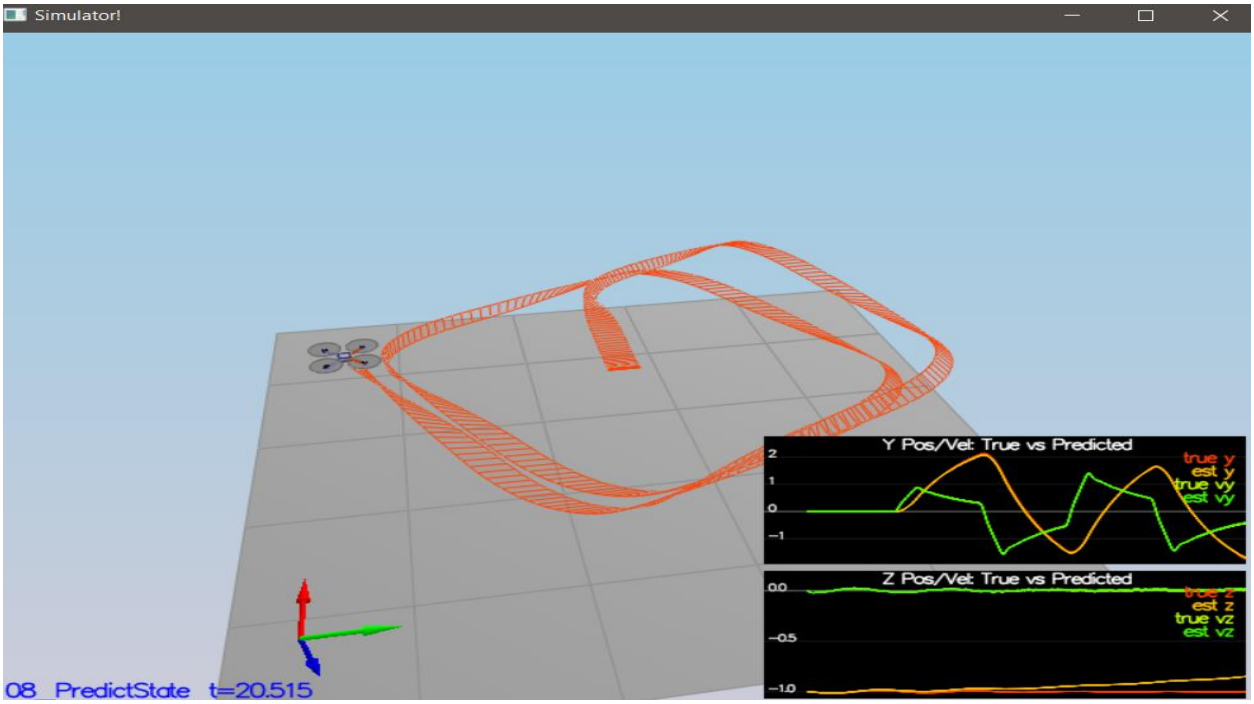
In this scenario we had to modify the UpdateFromImu() function. We needed to modify the linear implementation and implement a non-linear one in order to get the better results. The equation used was explained in the lecture it was to find derivatives of roll, pitch, yaw from angular rates in body frame p, q, and r.

The criteria for passing this task were to get attitude estimator within 0.1 rad for each of the Euler angles for 3 seconds. Below is the image of the same:



Prediction Step:

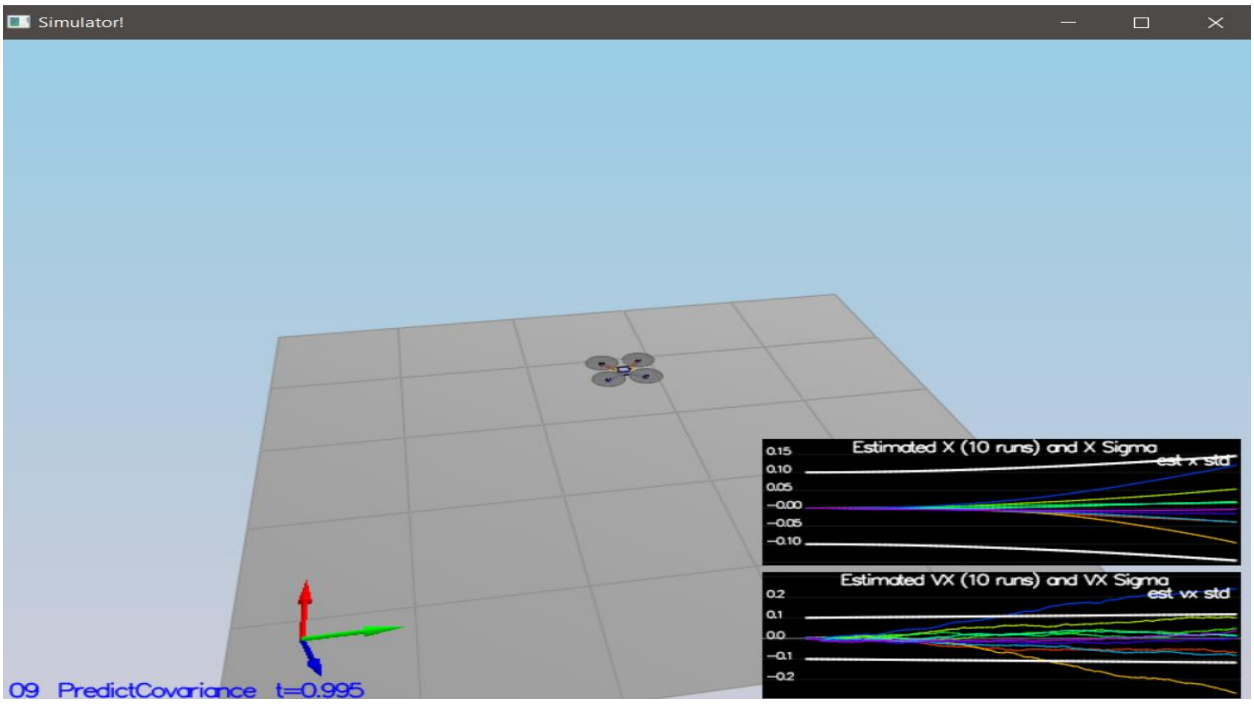
In this method we had to implement PreditctState() function. In order to predict the future state we used the current state and measurements from accelerometer and gyro. This scenario didn't have any specific criteria for passing. The simulation of the same is shown in below image:



Predict Covariance:

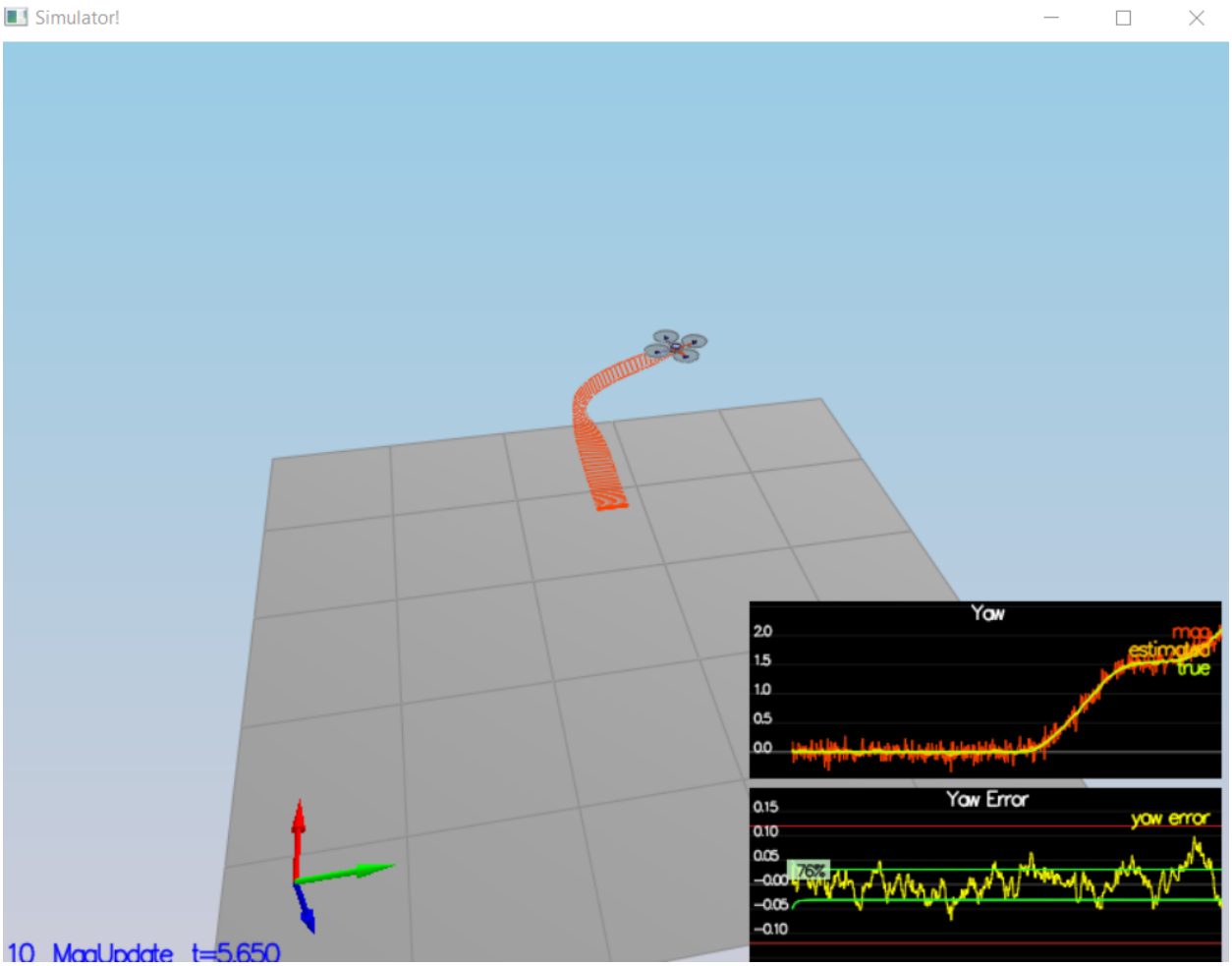
In this method we introduced an IMU with noise and observed the results and noticed that the estimated covariance matrix did not capture the growing errors, and we had to update the covariance matrix.

Below shown are the results:



Magnetometer Update:

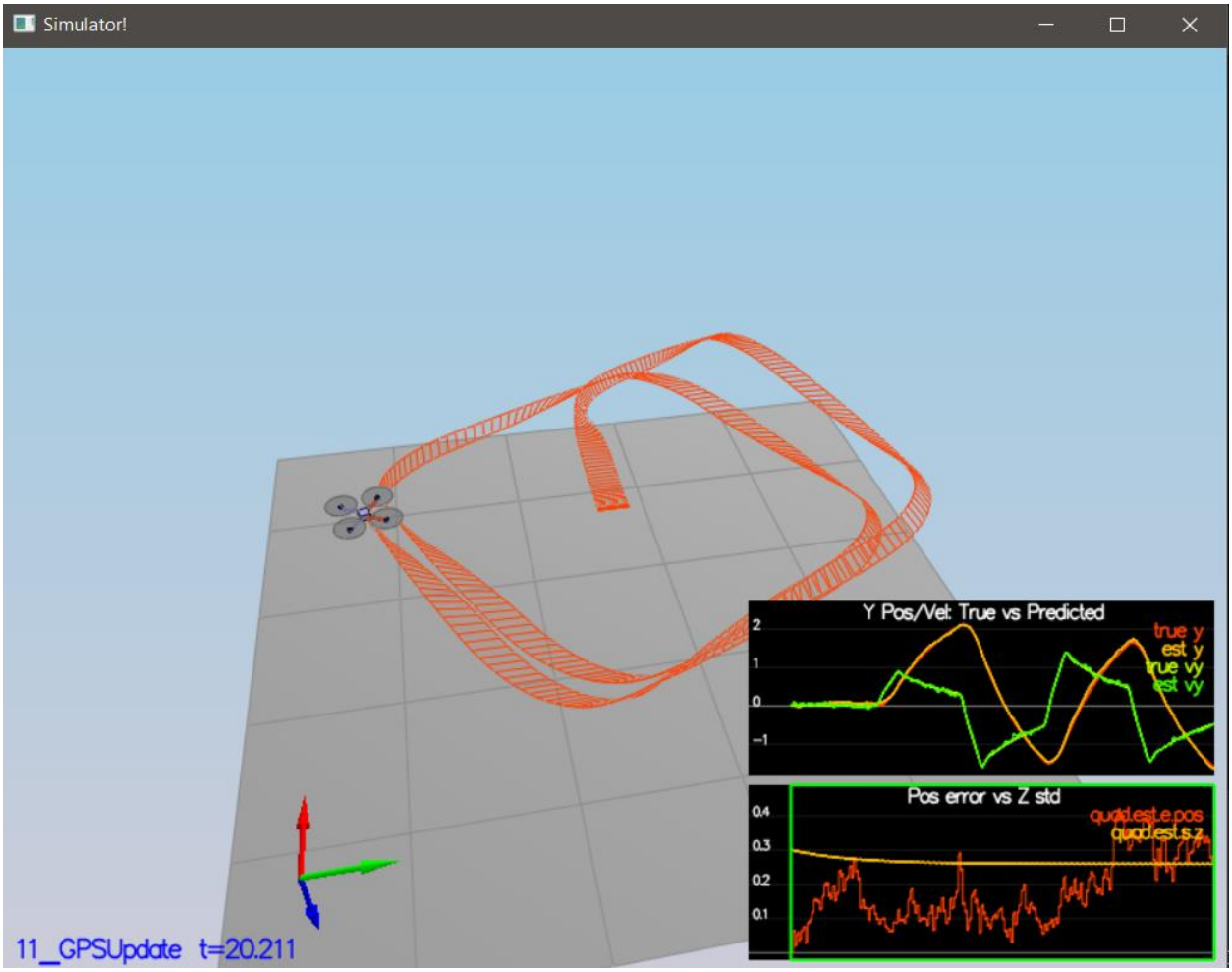
Now we use the magnetometer for our state estimation. We will implement the code for magnetometer in UpdateFromMag() function. After this we tuned the parameter QYawXY in order to get the magnitude of drift. The simulation of the same is shown below:



GPS Update:

In this step we now also incorporated the GPS measurement to complete our EKF implementation. The success criteria for this one was to complete the simulation cycle with an estimated position error of less than 1m.

Below is the image of simulation:


Simulator!
Y Pos/Vel: True vs Predicted
2
1
0
−1
true y
est y
true vy
est vy
Pos error vs Z std
0.4
0.3
0.2
0.1
quadest.e.pos
quadest.s.z
11_GPSUpdate t=20.211