Assignment 7: FLIGHT

Jockabeth Ponce

Group members: Ethan, Chris, Clay

ESE 448
Systems Lab

Introduction

The purpose of this assignment was to simulate the take off, hover for 1.5 seconds and landing of the UAV. We reused our simulink model of the linear and nonlinear dynamics of the UAV from an earlier assignment.

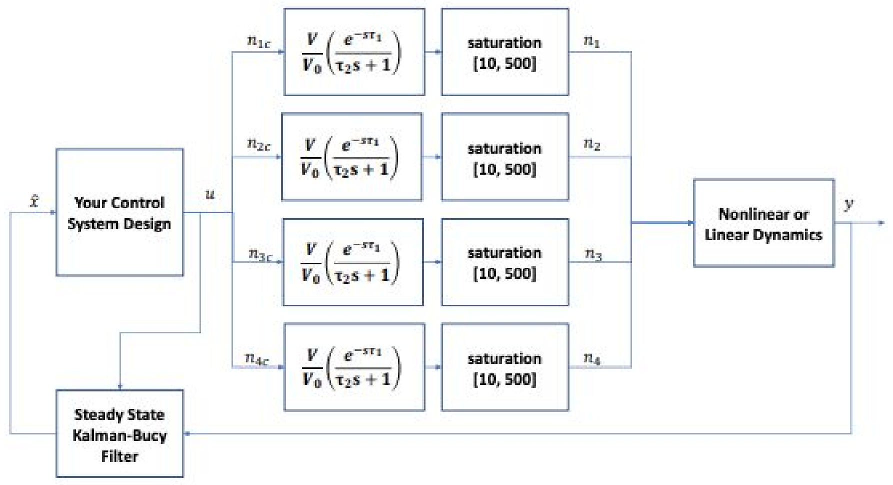
Some background information that was needed for this assignment was the transfer function of the actuator:

$$\frac{n_i}{n_{ic}} = \frac{V}{V_0}\left(\frac{e^{-\tau_1 s}}{\tau_2 s + 1}\right).$$

Professor Bhan Assignment 7

We were given the delay time T1 of 30ms and 10 Hz bandwidth which specifies the time constant T2 in the transfer function, this was calculated to be .0159. The actuator output is restricted to 10 and 500 Hz which is why there needs to be a saturation block before the output of the individual four motors.

The necessary components are shown below:



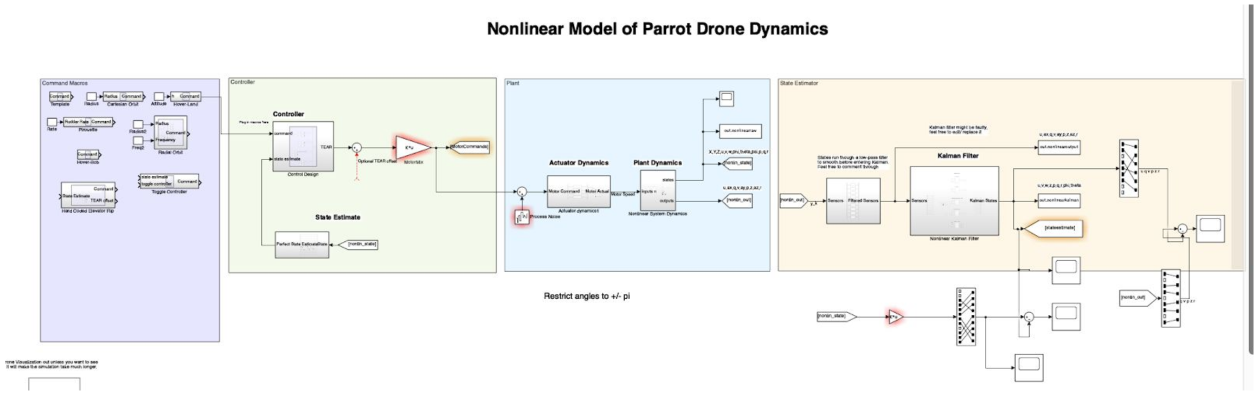
n_{1c}
n_{2c}
n_{3c}
n_{4c}
$\frac{V}{V_0}\left(\frac{e^{-s\tau_1}}{\tau_2 s+1}\right)$
saturation [10, 500]
n_1
n_2
n_3
n_4
$\hat{x}$
Your Control System Design
u
Nonlinear or Linear Dynamics
y
Steady State Kalman-Bucy Filter

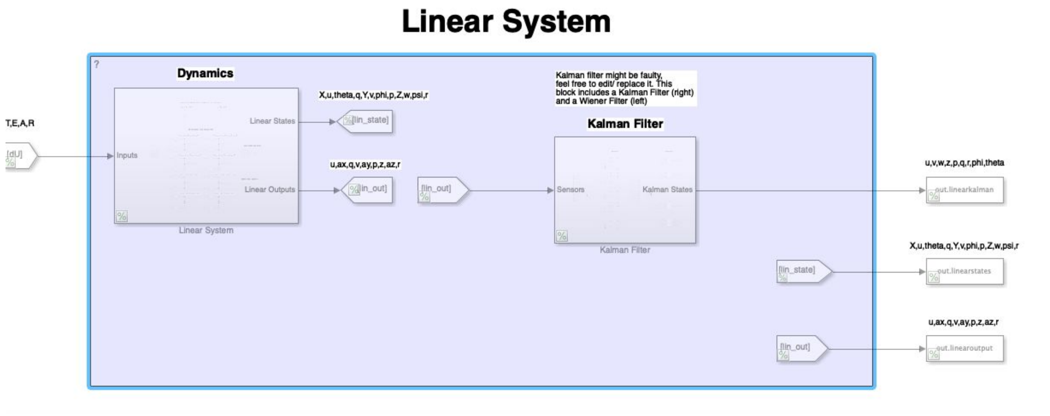
FIGURE 1. Simulation Architecture

Professor Bhan Assignment 7

The control systems vocabulary was difficult for me at first because I was not knowledgeable about what it means to have a controller, actuator, observer, etc in a model. This assignment clarified these components for me and put them to use. The model that made the most sense to me was Ethan's since I can see how all of these components came together. It was easy to read since the components were connected as if we were reading from left to right from Commands to linear and nonlinear outputs. (All models consisted of the same components, but Ethan's was easier for me to understand and we all switched to use the same model.)

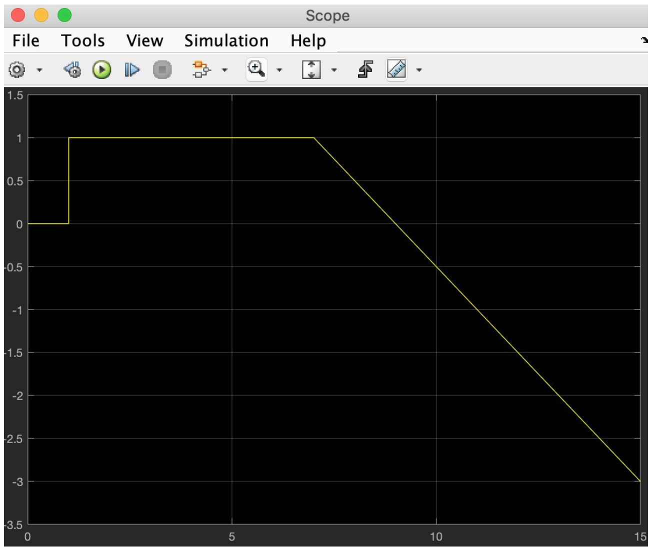
Simulink model:





Flight Control Simulation

In the hover-land command we multiplied a step times a ramp with a height of 1 (output of this is in the scope image below). Then we multiplied that output by the height we wanted, which was 1.5 m, and connected this to the z command, while all other commands (u, v, w, p, q, r, phi, and theta) were commanded to 0.

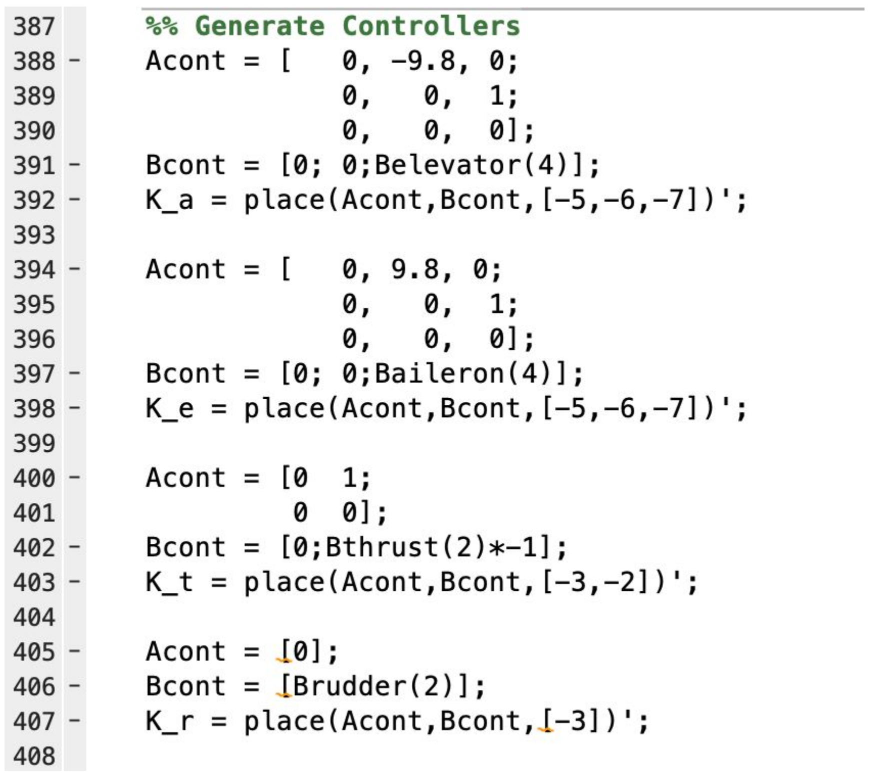


This output was multiplied by 1.5 so that the UAV will take off, hover at 1.5, then land.

In the controller, we separated the states that were related to T, E, A, and R respectively. To get the thrust controller we input the commands w and z, subtracted by the state estimates of w and z (this gives the error) and then multiplied by the Thrust controller gain.

To get the elevator controller we did the same thing with the states v, p, and phi. To get the aileron controller we did the same thing with the states u, q, and theta. Lastly, to get the rudder controller we did the same thing with the state r.

The matlab code to calculate the TEAR gains is shown below (K_t, K_e, K_a, K_r):



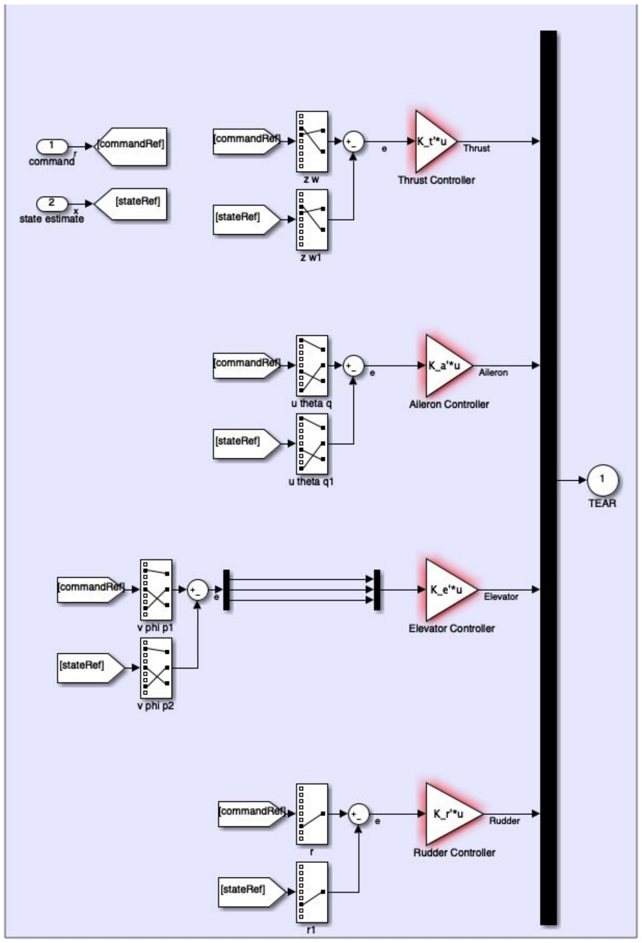
```
%% Generate Controllers
Acont = [   0, -9.8, 0;
            0,   0,  1;
            0,   0,  0];
Bcont = [0; 0;Belevator(4)];
K_a = place(Acont,Bcont,[-5,-6,-7])';

Acont = [   0, 9.8, 0;
            0,   0,  1;
            0,   0,  0];
Bcont = [0; 0;Baileron(4)];
K_e = place(Acont,Bcont,[-5,-6,-7])';

Acont = [0  1;
         0  0];
Bcont = [0;Bthrust(2)*-1];
K_t = place(Acont,Bcont,[-3,-2])';

Acont = [0];
Bcont = [Brudder(2)];
K_r = place(Acont,Bcont,[-3])';
```

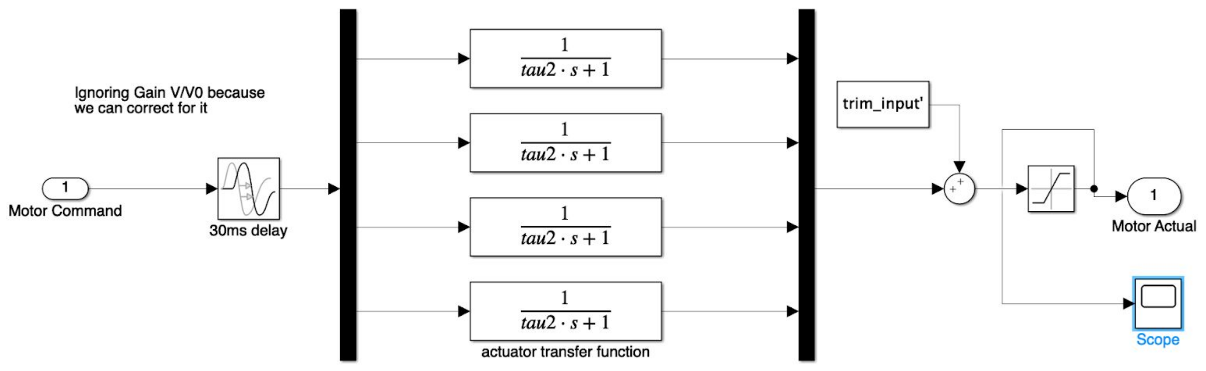
The controller simulink model is shown below:



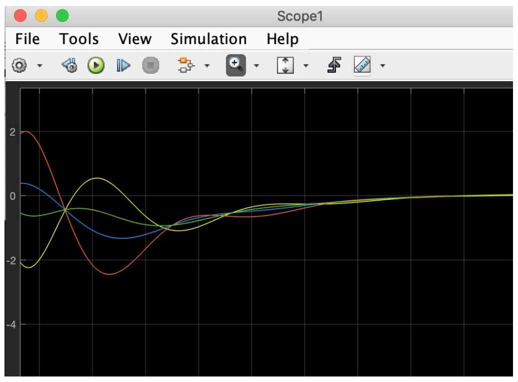
Controller in Simulink

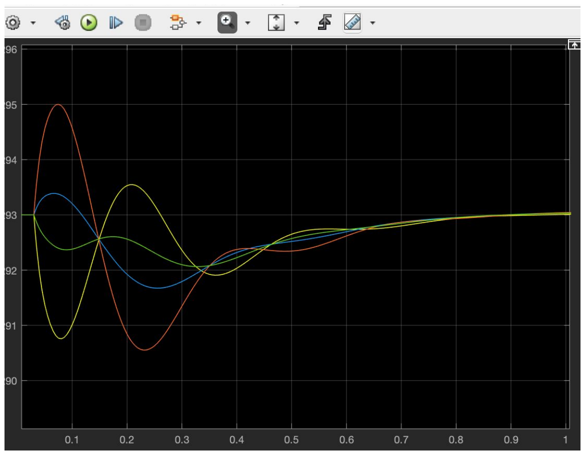
These 4 controllers were connected in a mux to create TEAR which was then connected to the motor mixing matrix and output the motor commands.

These motor commands were input to the actuator dynamics:


Ignoring Gain V/V0 because we can correct for it
1
Motor Command
30ms delay
$\frac{1}{tau2 \cdot s + 1}$
actuator transfer function
trim_input'
1
Motor Actual
Scope

Instead of having the e^{-sT} in the numerator as shown in the introduction, we added a 30ms delay using a Transport Delay simulink block. The trim input used was 293 Hz so that all motors can stabilize to this value. Tau2 was calculated to be .0159.

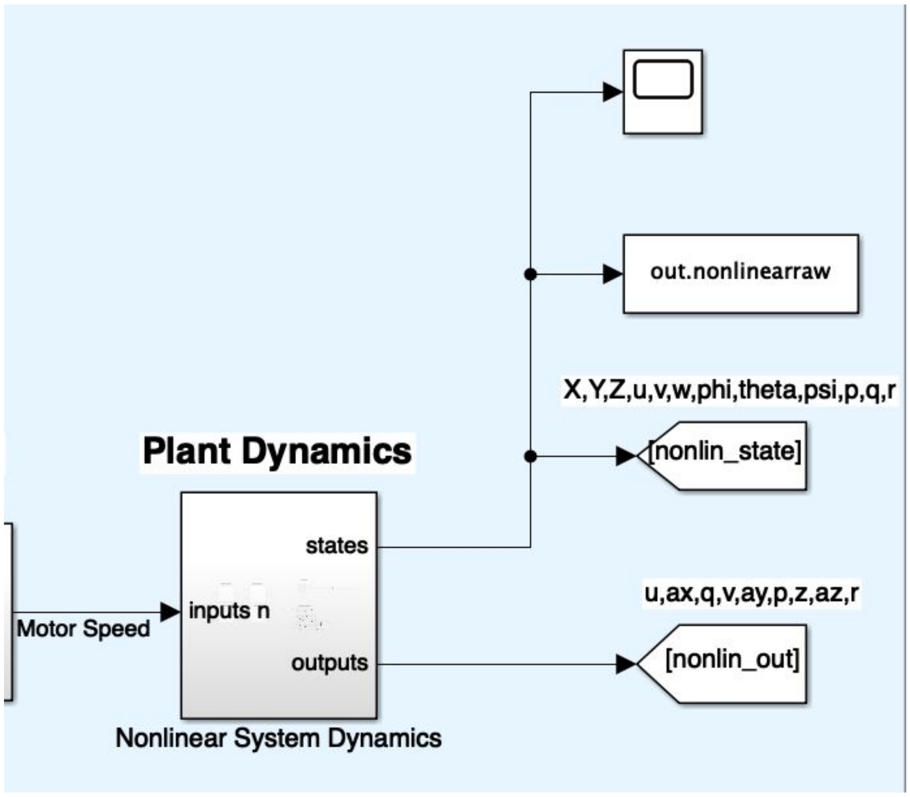




Moto outputs before adding trim

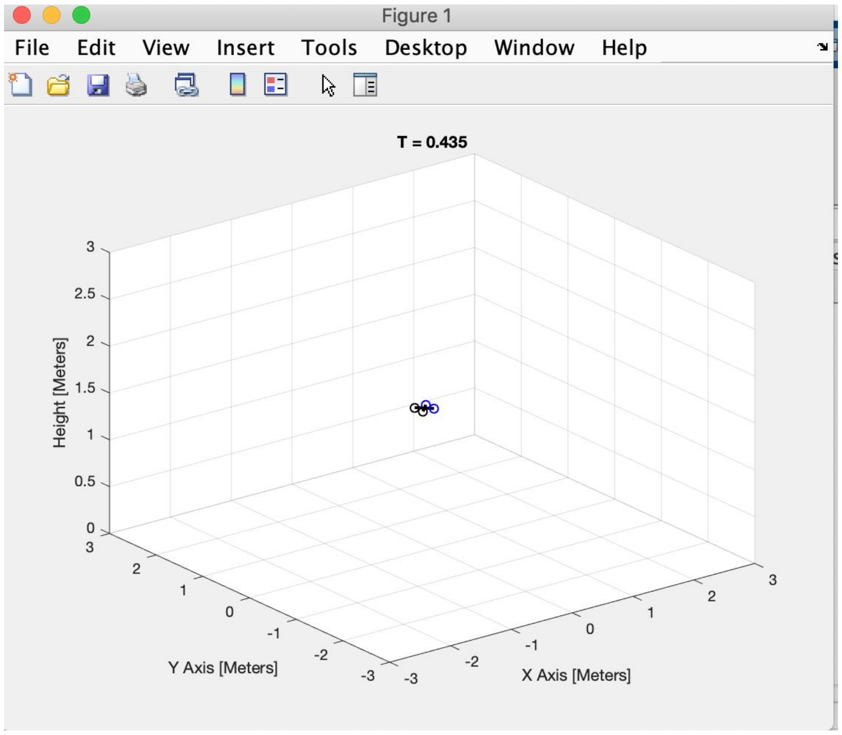
Motor outputs after adding trim

These motor speeds were input into nonlinear system dynamics.



Nonlinear System Dynamics

Using the to workspace block we can use the DroneVisualizer.m code to simulate the drone in matlab. This code was provided by TA Terence Havlik, the only states needed to use the drone visualizer code are Xned, Yned, Zned, theta, phi, and psi in that order.



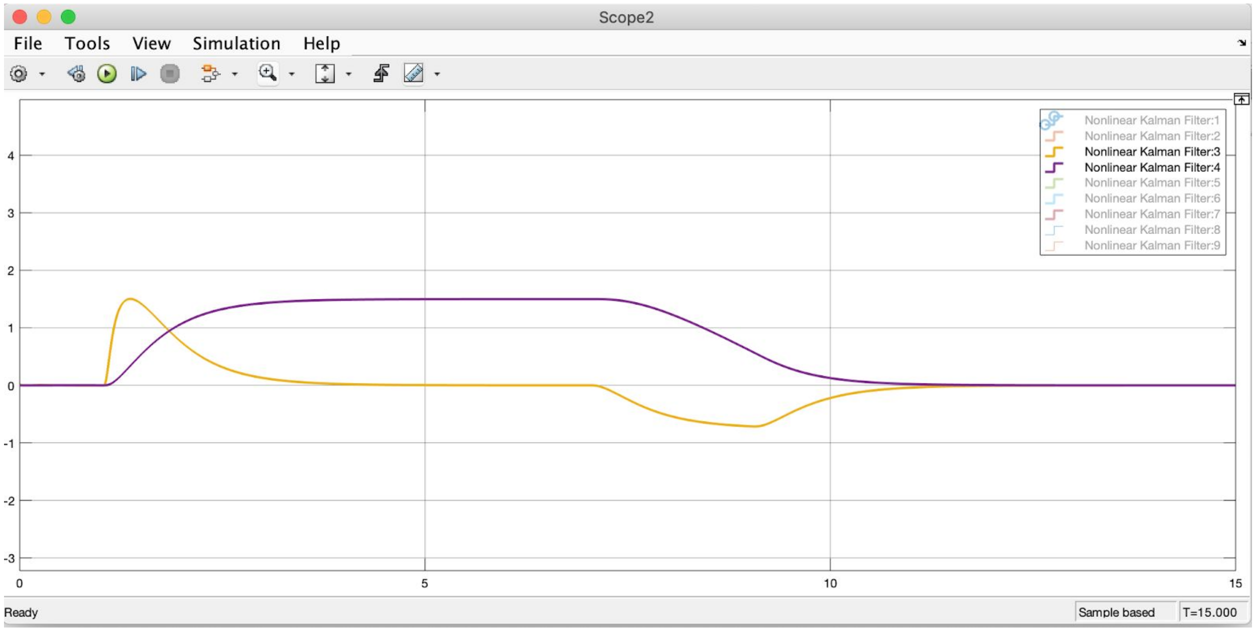
Link of simulation of hover I uploaded into wustl box:

https://wustl.box.com/s/q1eeb3cwzou91iy6ea6mbrityoeomt5e

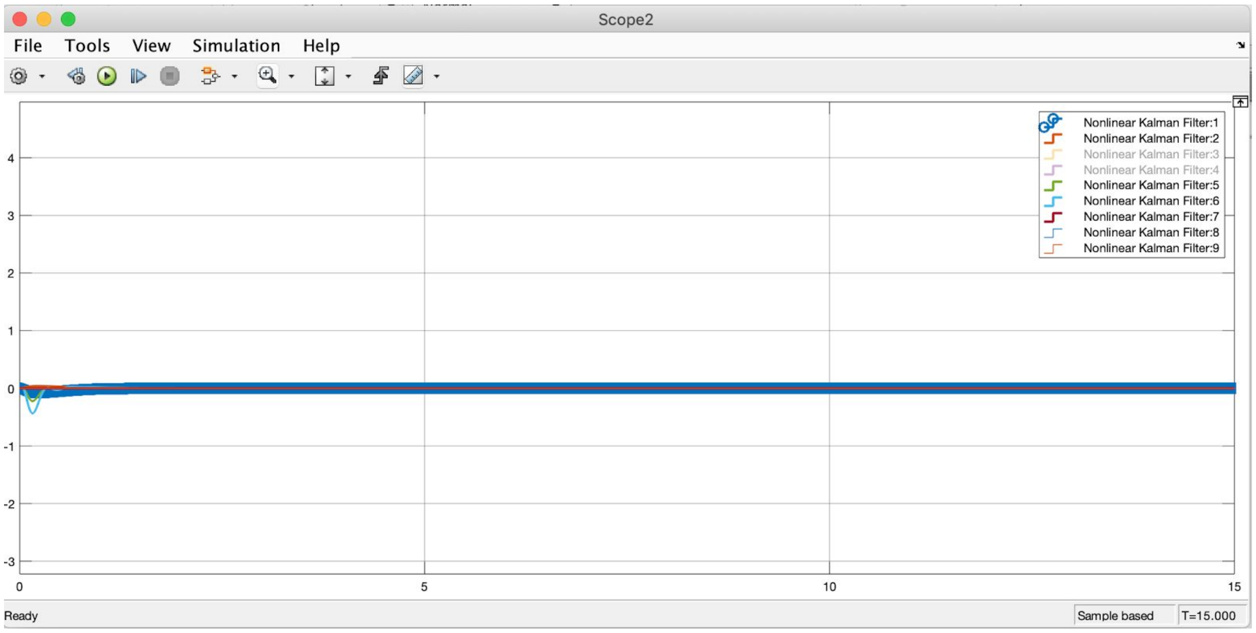
This was very exciting for our group the first time we saw our simulated drone fly!

The nonlinear outputs from the nonlinear system dynamics are then put into a low pass filter to smooth out the signals before entering the Kalman filters.

W and Z were the 3rd and 4th outputs of the Kalman estimates and the plot of the take off, hover at 1.5 m, and landing is shown below:



All other states were commanded to zero:



Conclusion

This assignment was very rewarding because we saw all of the pieces come together to finally see our *simulated* drone fly. The drone visualizer was excellent because at least we can see that our simulation worked even though we could not get the physical drone in the air. Hovering was such an accomplishment for our group and it was very exciting.