Assignment 7 ESE 448

Clay Whipp

4/23/20

In this assignment, I connected four components together: the linear and nonlinear dynamics, a Wiener filter, a controller, and the actuator dynamics. My objective was to use these components in a closed-loop system to command the drone in simulation to begin on the ground, rise to a 1.5m hover, and then land. A diagram of the system is shown below.

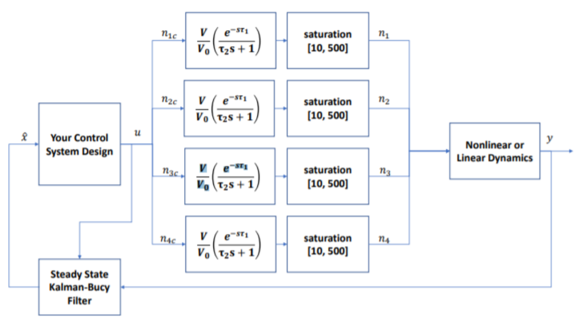

Figure 1: Closed-loop system used to control states

1 Piecing Together the System

Since I already knew what the structure of the system was, I just had to build it piece by piece. The first step was to implement both the linear and non-linear dynamics. I discovered that my non-linear dynamics were not correct after testing a basic throttle input. This means that my linear dynamics were wrong too since both the linear and non-linear dynamics came from the same equations in Matlab. To remedy this, I used Chris' linear and non-linear dynamics.

2 Testing the Dynamics

I tested the dynamics by inputting a step function for a throttle, elevator, aileron, rudder input separately and got the correct results. The step input started at 0, then jumped to 1 at 1 second and remained there until 10 seconds.

Here is the Simulink diagram I built to test the dynamics.

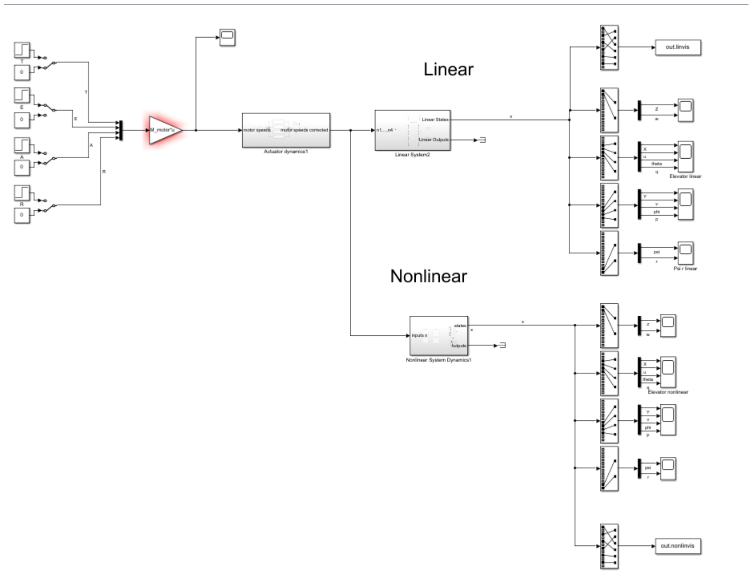

Figure 2: Simulink diagram to test dynamics

Here are the outputs of the linear dynamics for T, E, A, and R inputs.

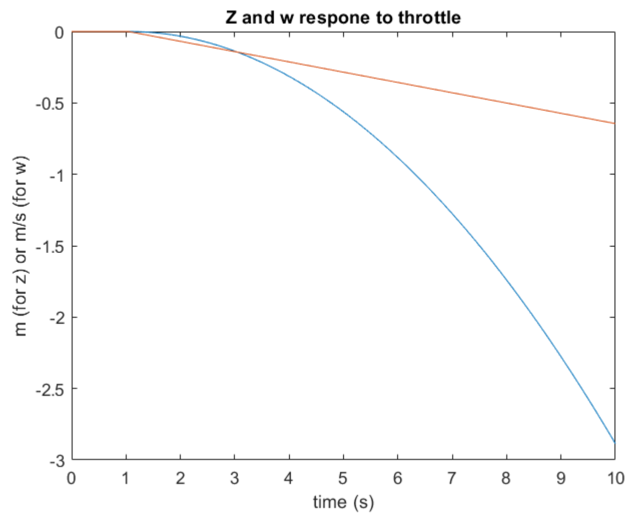

Figure 3: Linear Z and w response to throttle

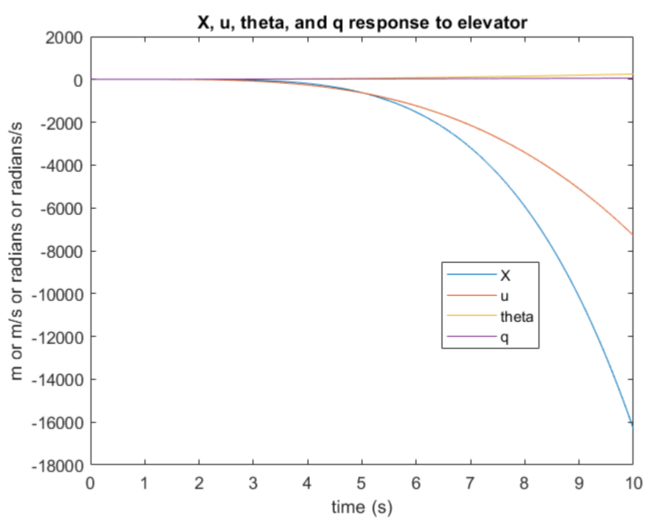

Figure 4: Linear X, u, theta, and q response to elevator

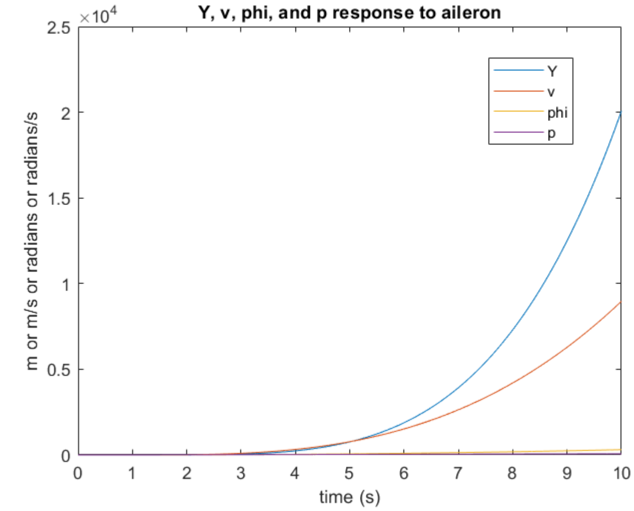

Figure 5: Linear Y, v, phi, and p response to aileron

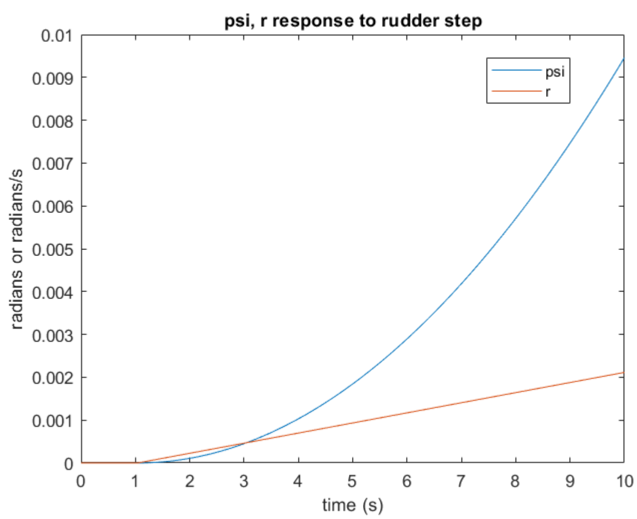

Figure 6: Linear psi, and r response to rudder step

Here are the outputs of the nonlinear dynamics for T, E, A, and R inputs.

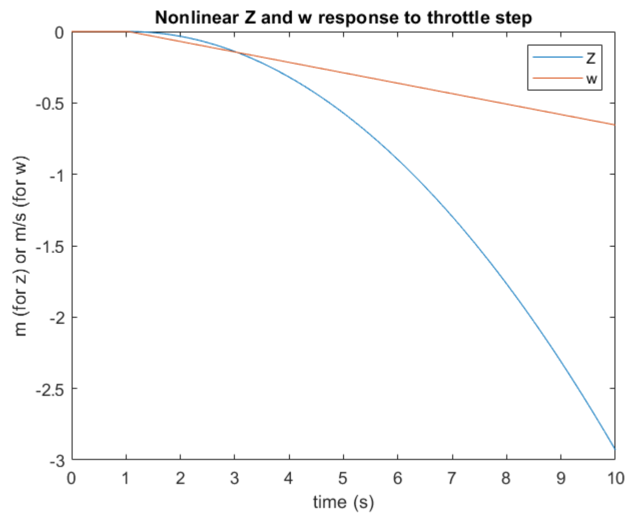

Figure 7: Nonlinear Z and w response to throttle step

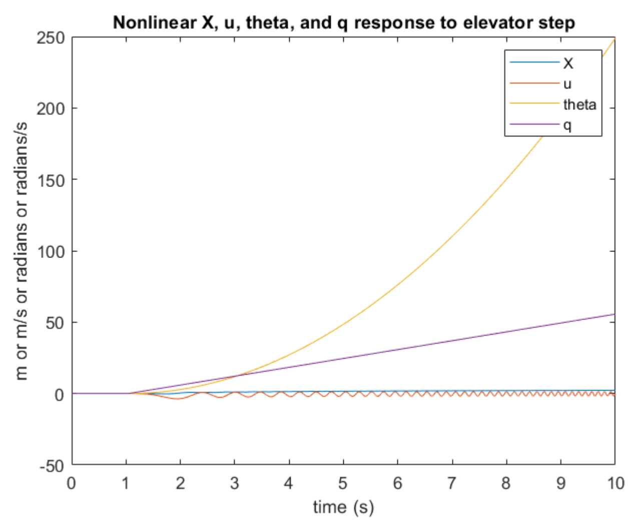

Figure 8: Nonlinear X, u, theta, and q response to elevator step

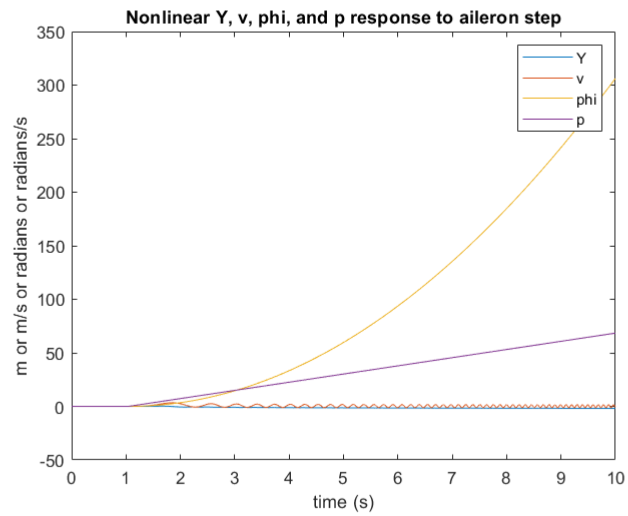

Figure 9: Nonlinear Y, v, phi, and p response to aileron step

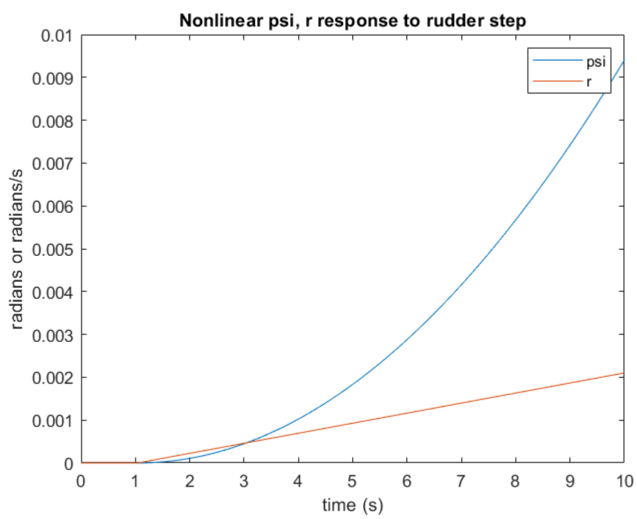

Figure 10: Nonlinear psi, and r response to rudder step

3 Connecting the Wiener filter

For the filter, I simple connected the dynamics to my Z Wiener filter. Currently this is the only Wiener filter I am using, but in the future I would like to implement the filters for the states we can't measure in order to run this on the actual drone. The Simulink for my Z wiener filter is shown below.

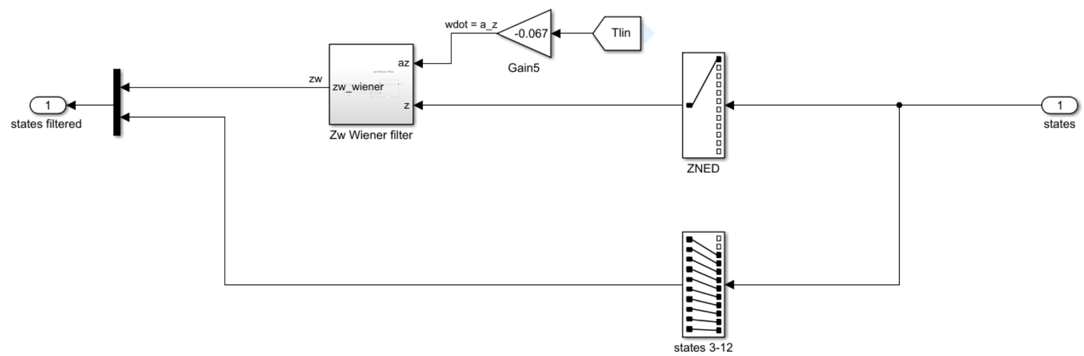

Figure 11: Simulink of the Z Wiener filter

4 Building the Controller

To make the controller I used the LQR function in Matlab. Currently I am only used proportional or the "P" in PID for my controller. In the future if I need to I can also add integral or "I" making my controller a PI controller.

The LQR function gave my my gains, K for T, E, A, and R. Using these I take the error, $e = x_{desired} - x_{actual}$ and multiply by K to get $u = Ke$. Here is the Simulink diagram for my controller. Everything gets set to 0 except position which gets set to 0, then -1.5 (since h = -z) then 0 again.

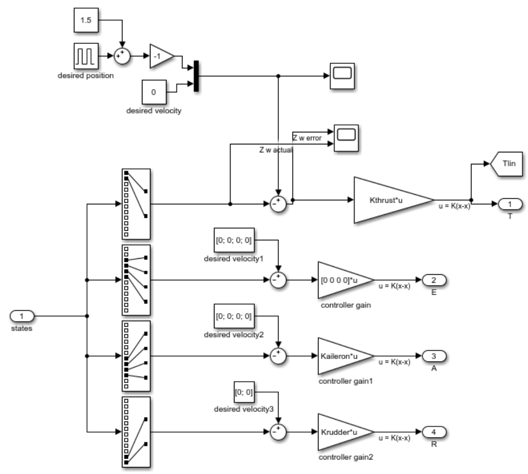

Figure 12: Simulink of the LQR Controllers

5 Actuator Dynamics

The actuator dynamics were the final piece of the puzzle. I used the homework background to find the basic structure of the actuator's transfer function which is shown below.

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

Figure 13: Actuator transfer function

I ignored $\frac{V}{V_0}$ since I can just multiply by $\frac{V_0}{V}$ to cancel out whatever error is being caused by the voltage. τ_1 was given to us as $\tau_1 = 30$ms. To find τ_2, I used the estimate we were given that the actuator model has a bandwidth of 10Hz. This means that the gain of the transfer function decreases by a factor of $\frac{1}{\sqrt{2}}$ at 10Hz. Using this I solved for $\tau_2 = 0.0159$. I added a saturation block and connected the actuator dynamics back to the dynamics, completing the loop. I subtracted, n_{trim}, which is the speed of the motors in order to achieve 1g trim from the motor speeds going into the linear system, because we linearized the dynamics around 1g trim.

6 Take off, hover, and landing

Here is the Simulink for the closed-loop system.

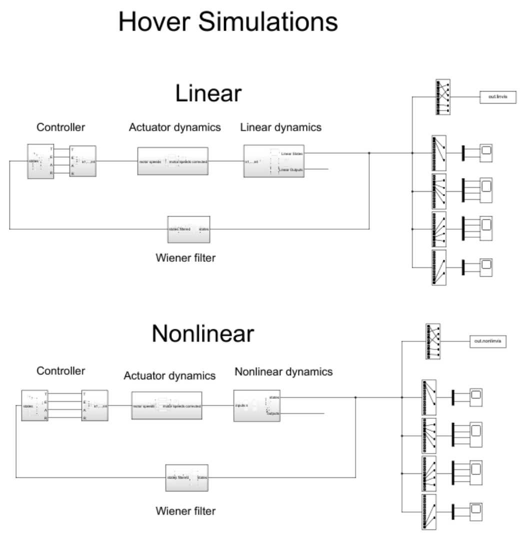

Figure 14: Simulink of closed-loop system

The whole goal of this homework was to use the controller along with the rest of the closed-loop system to have the drone take off, hover at 1.5m and land. Here is the desired Z and w states I inputted into the controller.

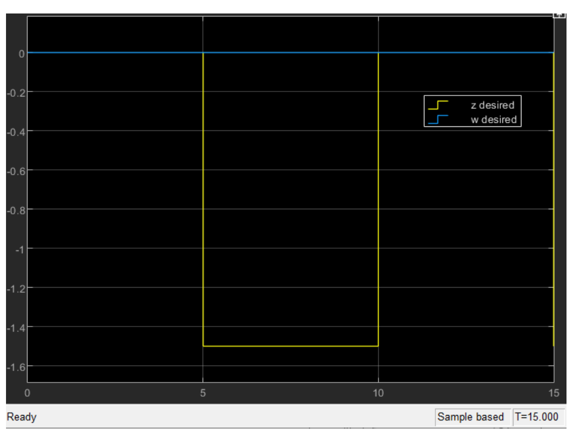

Figure 15: Z and w desired

Here is the actual Z and w states responding to this input for both the linear and nonlinear systems.

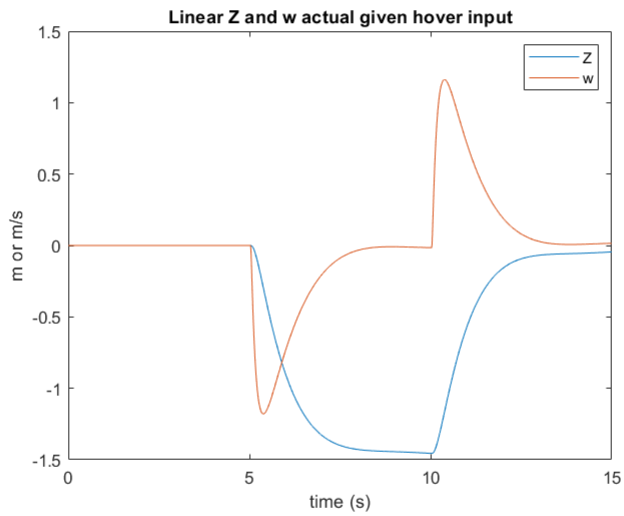

Figure 16: Linear Z and w actual given hover input

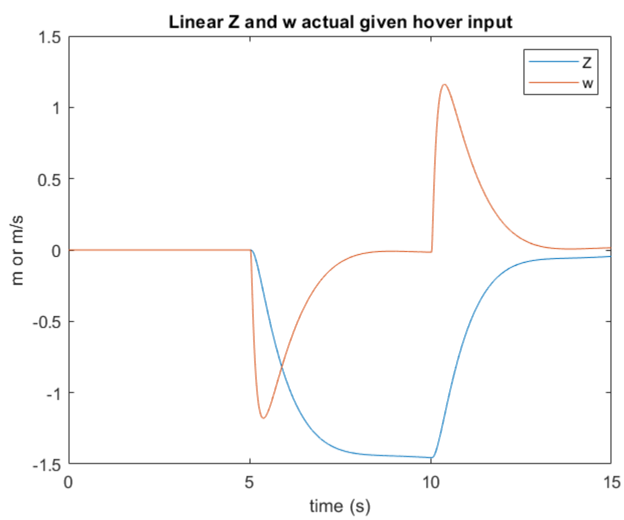

Figure 17: Nonlinear Z and w actual given hover input

Both the linear and the nonlinear Z and w states look great. As commanded, Z starts at 0, go to -1.5 at 5 seconds, then back to 0 at 10 seconds. Part of the reason the responses are so smooth is there is no noise in the system, the only "imperfections" are the actuator dynamics and the one Wiener filter. To make this more realistic I could add some white noise into the Wiener filter or implement more Wiener filters for the signals that we can't measure, such as ϕ

and θ.

7 Conclusion

In summary, I am very happy with my results for this assignment. All of the Matlab and Simulink I have been working on throughout the semester came together and I was able to sucessfully control the drone in simulation. If I had more time I would move on to controlling the actual drone.