Introduction to Control and Robotic Systems

EE-386

Mini-Project

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Hunter Wilhelm Phillips

February 6, 2019

GitHub Repository at

https://github.com/robolux/EE_Controls

1. Introduction

The base of this project lies in simulation of the pitch-plane rocket stabilization problem using an appropriately designed control system in Matlab Simulink. The goal of the controller is to maintain an angle of attack equal to zero, in simple terms, a vertical flight path. It is established that the controller must handle a simulated wind disturbance in the form of a step function. There will be several simulations undertaken including first an uncompensated plant to provide a reference for an uncontrolled system. Second a Proportional Derivative (PD) controller will be simulated and finally a Proportional Integral Derivative (PID) controller to round out the variety of testing. The extensive testing to be completed will provide a basis for comparison between each control system and its associated drawbacks and advantages. First the mathematical model will be formulated and the problem statement outlined in preparation for the controller design stage. With this established the control systems will be designed and implemented in Simulink to perform the necessary simulations and associated graphical data. The results of the runs will be analyzed and formed into a conclusion concerning the scope of the mini-project.

2. Mathematical Modeling and Problem Formation

2.1 System Decomposition

To first characterize the system under evaluation the rocket free body control diagram must be established. The drawn out free body diagram can be see below in *Figure 1* as found in the coursework [1].

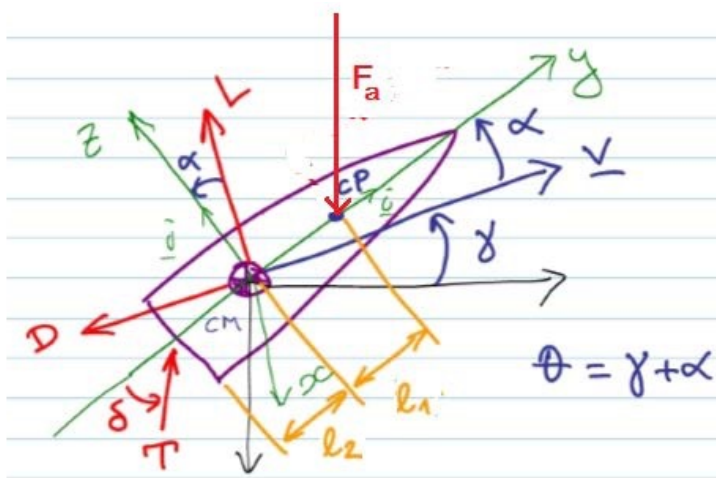

Figure 1: Free Body Diagram of Rocket

For this project, the end goal is to drive the angle of attack (α) to 0°, which is also coupled with the flight path (γ) being maintained at 90°. To make the simulation easier to resolve, the engine thrust (T) and aerodynamic forces (F_A) were assumed to be constant values. Since the general simulation goal has been set forth, the functional diagram of the control algorithm can be seen below in *Figure 2*.

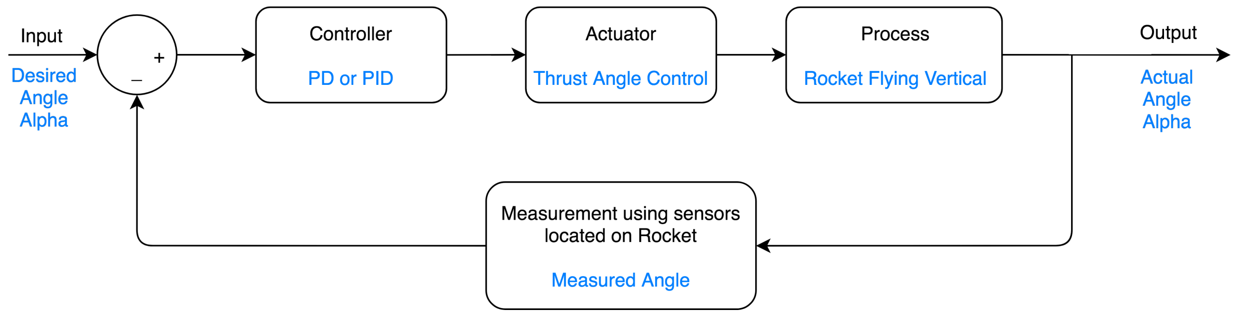

Figure 2: Flow Diagram of Generalized Control Algorithm with Mini-Project specific characterizations highlighted in blue.

2.2 Linearization

To mathematically model the system, the problem must be linearized for simplicity with relation to the knowledge obtained during the class. This can be started by obtaining (1) as seen below:

$$J\varepsilon = \sum_i T_i \rightarrow J\ddot{a} = Tl_2 \sin\delta + F_a l_1 \sin\alpha \tag{1}$$

coupled with $\alpha_{eq} = \delta_{eq} = 0$

Where α_{eq} and δ_{eq} are the angles at equilibrium

By then taking the partial with respect to the function the decomposition can begin to occur as seen in (2).

$$f(x,y) \approx \mathrm{f}(x_{eq}, y_{eq}) + \frac{\partial f(x_{eq}, y_{eq})}{\partial x}(x - x_{eq}) + \frac{\partial f(x_{eq}, y_{eq})}{\partial y}(y - y_{eq}) \tag{2}$$

Where x_{eq} and y_{eq} are the states at equilibrium

To ensure that the problem remains linearized, the bounds must be kept as defined below in the set of equations (3).

$$\sin(\delta) \approx \sin(0) + \cos(0)\,(\delta - 0) = \delta, \text{if } |\delta| \leq \frac{\pi}{6}$$
$$\sin(\alpha) \approx \sin(0) + \cos(0)\,(\alpha - 0) = \alpha, \text{if } |\alpha| \leq \frac{\pi}{6} \tag{3}$$

To further simplify our equations under test, the (1) can be rewritten as (4).

$$\ddot{\alpha} = \frac{Tl_2}{J}\delta + \frac{F_a l_1}{J}\alpha$$
$$\ddot{\alpha} = b_1\delta + a_1\alpha \tag{4}$$

where a_1 and b_1 are the reduced constants

The derived equation, (4) is the finalized mathematical model of the plant system under consideration. In this project, $b_1 = 14$ and $a_1 = 5$ are given in the prompt and will be used in the appropriate sections in the future model. If the input and output of the system are bounded, the system will be bounded input – bounded output (BIBO) stable with respect to a stable linearization. This allows for a fluid transition into defining the appropriate stages of the simulation with the respective control algorithm being used for each.

2.3 Uncompensated Rocket Equation

To first set forth a baseline to test our controller standards against, an uncompensated "controller" was designed to simulate a boundless situation. This takes the form of a simple representation of (4) formed into (5) as seen below:

$$\ddot{\alpha} - a_1\alpha = 0 \tag{5}$$

It can be observed that this will not go well for the rockets trajectory since the form of (5) does not include any derivation or integration. This should result in a vastly unstable system incapable of vertical flight.

2.4 Proportional Derivative (PD) Controller

The decision to use a PD controller is key since it has incredibly fast response time which is a major plus in control systems. This however comes with the drawback of not having the best response to errors occurring in the system from external disturbances. The first step in developing the PD controller is to stabilize the systems output at zero such that (6) and (7) are introduced from [2]:

$$\lim_{t \to \infty} \alpha(t) = 0 \tag{6}$$

$$\delta = k_1 e_\alpha + k_2 \dot{e_\alpha} = -k_1 \alpha - k_2 \dot{\alpha} \tag{7}$$

where k_1 and k_2 are the distinct gains

The system dynamics take the compensated form seen in the coupled set of equations (8) and (9) using (7) substituted into a rewritten (4):

$$\ddot{\alpha} - a_1 \alpha = b_1 \delta \tag{8}$$

$$\ddot{\alpha} + b_1 k_2 \dot{\alpha} + (b_1 k_1 - a_1)\alpha = 0 \tag{9}$$

By taking (9) which is in the form of a charateristic equation, it can be seen that it is charaterized by (10) below:

$$\lambda^2 + r_2 \lambda + r_1 = 0 \tag{10}$$

Using the quadratic formula, (11) can be observed, which is asymptotically stable with the current states of (12).

$$\alpha(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} \tag{11}$$

$$r_1, r_2 > 0 \text{ since } Re(\lambda_{1,2}) < 0 \tag{12}$$

Taking the traditional control form of (13) with associated variables being assigned in (14):

$$\ddot{\alpha} + 2\xi\omega_n \dot{\alpha} + \omega_n^2 \alpha = 0 \tag{13}$$

$$t_s = \frac{4}{\xi\omega_n},\ \omega_n = \frac{4}{\xi t_s} \tag{14}$$

With ξ as the damping factor and ω_n as the natural frequency

Using this to derive the PD controller gains seen in the set of equations (15):

$$k_1 = \frac{a_1+\omega_n^2}{b_1}, k_2 = \frac{2\xi\omega_n}{b_1} \tag{15}$$

Plugging (15) synced with (14) into (7) forms the critical equation for the PD control system, (16):

$$\delta = -\left(\frac{\alpha_1}{b_1} + \frac{16}{(\xi t_s)^2 b_1}\right)\alpha - \left(\frac{8}{t_s b_1}\right)\dot{\alpha} \tag{16}$$

Using (16) to pertain to our case at hand with values given by the prompt, (17) can be finally reduced:

$$a_1 = 5,\ \ b_1 = 14,\ \ \xi = 0.7, and\ t_s = 1.5 \text{ seconds}$$
$$\delta = -1.394\alpha - 0.381\dot{\alpha} \tag{17}$$

This final form, (16), sets up for the final stretch of defining the PD controller, now it is time to prepare it for wind disturbances, $w(t)$. First, stating the coupled set of equations, (18) can be setup:

$$\ddot{\alpha} - a_1\alpha = b_1\delta + w(t)$$
$$\delta = -k_1\alpha - k_2\dot{\alpha} \tag{18}$$

From this (19) can be obtained:

$$\ddot{\alpha} - a_1\alpha = b_1(-k_1\alpha - k_2\dot{\alpha}) + w(t)$$
$$\ddot{\alpha} + b_1k_2\dot{\alpha} + (b_1k_1 - a_1)\alpha = w(t) \tag{19}$$

Then plugging in for the coefficients, b_1k_2 and $b_1k_1 - a_1$, it becomes (20).

$$\ddot{\alpha} + r_2\dot{\alpha} + r_1\alpha = w(t)$$
$$\text{with } r_1, r_2 > 0 \tag{20}$$

This equation is in the form of a nonhomogenous differential equation and is BIBO stable as prescribed, but the limit as time approaches infinity for α is not zero as conventionally accepted for the states defined.

2.5 Proportional Integral Derivative (PID) Controller

The main considering factor when implementing a PID controller is the huge advantage it has when overcoming disturbances. This comes at a slight computational cost, but with real world systems, this is advantageous due to unforeseen errors occurring in testing environments. The core equation when designing a PID controller is in the form of (21) from [2].

$$\delta = k_1 e_\alpha + k_2 \dot{e_\alpha} + k_3 \int e_\alpha dt \quad \text{with } e_\alpha = \alpha_c - \alpha \tag{21}$$

This is correct to method and drives the limit of α to zero as anticipated. Using a lot of the equation states as expansion references from PD, the following set of equations, (22), can be obtained neatly:

$$\begin{cases} \ddot{\alpha} - a_1 \alpha = b_1 \delta + w(t) \\ \delta = -k_1 \alpha - k_2 \dot{\alpha} - k_3 \int \alpha \, dt \end{cases} \tag{22}$$

Driving this combination seen in (22) into a final differential form of (23):

$$\begin{gathered} \ddot{\alpha} + b_1 k_2 \dot{\alpha} + (b_1 k_1 - a_1)\alpha + b_1 k_3 \int \alpha \, dt = w(t) \\ \dddot{\alpha} + r_3 \ddot{\alpha} + r_2 \dot{\alpha} + r_1 \alpha = 0 \end{gathered} \tag{23}$$

Defining the standard quadratic set as seen in (10) through (13), the stability conditions in (24) form into (25):

$$r_1, r_2, r_3 > 0, r_2 r_3 > r_1 \tag{24}$$

$$\dddot{\alpha} + 1.75\omega_n \ddot{\alpha} + 2.15\omega_n^2 \dot{\alpha} + \omega_n^2 \alpha = 0 \tag{25}$$

Defining the remaining variables needed to reduce to a final solution can be seen in (26):

$$t_s \approx \frac{10}{\omega_n} \to \omega_n \approx \frac{10}{t_s} \tag{26}$$

Setting up the final PID controller gains as follows in (27):

$$k_1 = \frac{2.15\omega_n^2 + a_1}{b_1}, k_2 = \frac{1.75\omega_n}{b_1}, k_3 = \frac{\omega_n^3}{b_1} \tag{27}$$

Finally using (22) with (27), the final differential form can be achieved as showcased in (28):

$$\delta = -\left(\frac{2.16 * 100}{b_1 t_s^2} + \frac{a_1}{b_1}\right)\alpha - \left(\frac{1.75 * 10}{b_1 t_s}\right)\dot{\alpha} - \left(\frac{1000}{t_s^3 b_1}\right)\int \alpha\, dt \tag{28}$$

Finally applying this to the case at hand, (29) is solved using the given variables:

$$a_1 = 5,\ b_1 = 14,\ and\ t_s = 1.5$$

$$\delta = -7.214\alpha - 0.833\dot{\alpha} - 21.164\int \alpha\, dt \tag{29}$$

Remembering the form of (23) the complete mathematical derivation has occurred including the acceptance of wind disturbances.

3. Discussion of Controller Design

3.1 Uncompensated Plant

This is the most trivial of the setups in Simulink. There are just two basic integrators with the corresponding gain being applied. It is fairly obvious that this will quickly spiral exponentially out of control. The math flow diagram can be seen in *Figure 3* below.

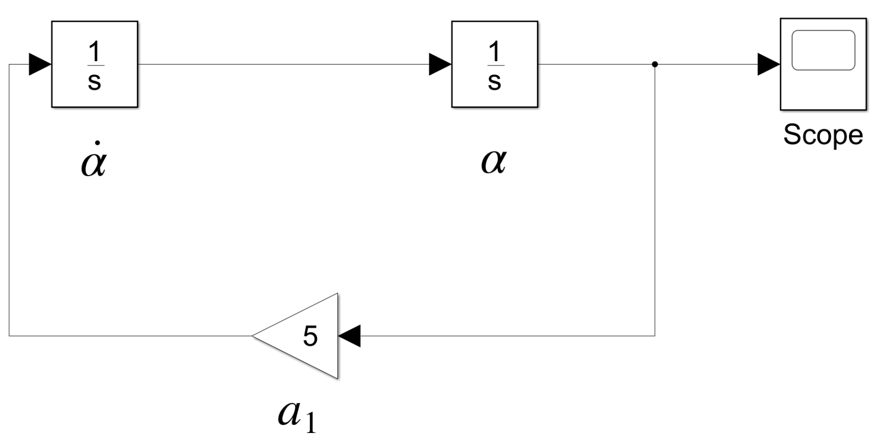

Figure 3: Simulink Block Diagram Model of the Uncompensated Plant

3.2 PD Controller with No Disturbance

This is the stereotypical use for a PD controller and provides an ideal situation for the testing to occur. There is no disturbance occurring in the form of $w(t) \equiv 0$. The math flow diagram can be viewed in *Figure 4* below.

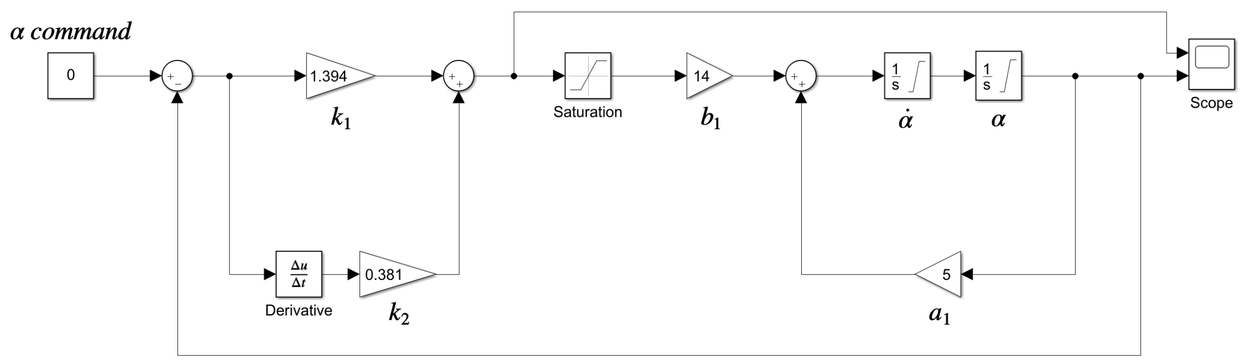

Figure 4: Simulink Block Diagram Model of the PD Controller with No Disturbance

3.3 PD Controller with Disturbance

This is a use case where a PD controller does not handle the situation well, since it might be fast, but cannot handle error within a simulation solution. By having this counteracting point against the previous simulation, more comparisons can be drawn between controller designs. The math flow diagram can be observed in *Figure 5* below.

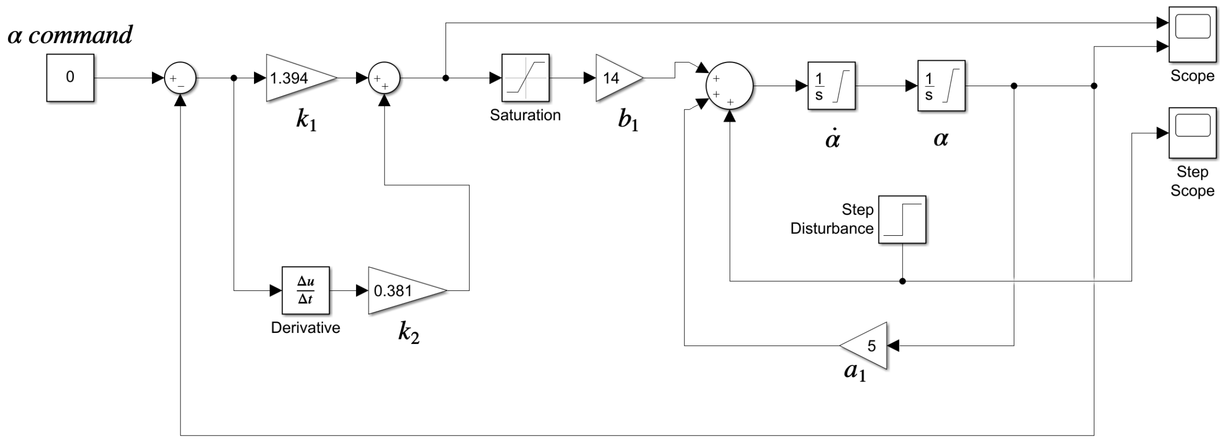

Figure 5: Simulink Block Diagram Model of the PD Controller with Disturbance

3.4 PID Controller with Disturbance

This is the projected optimal solver for a system with disturbance as it can handle the error in the solution well. It however comes at a slight cost with performance time, but with modern computing solutions has become heavily prevalent in control systems. This is true for the use-case at hand, keeping a rocket in vertical flight with external wind disturbances. The math flow diagram can be seen in *Figure 6* below.

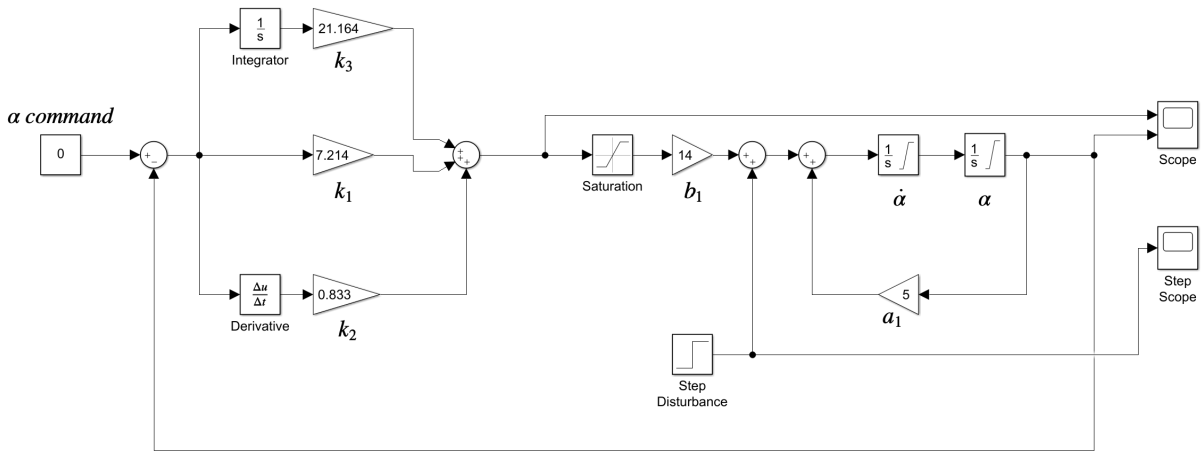

Figure 6: Simulink Block Diagram Model of the PID Controller with Disturbance

4. Simulations

4.1 Uncompensated Plant Run

The first simulation ran, used the uncompensated controller found in **Section 3.1**, with the results shown in *Figure 7*.

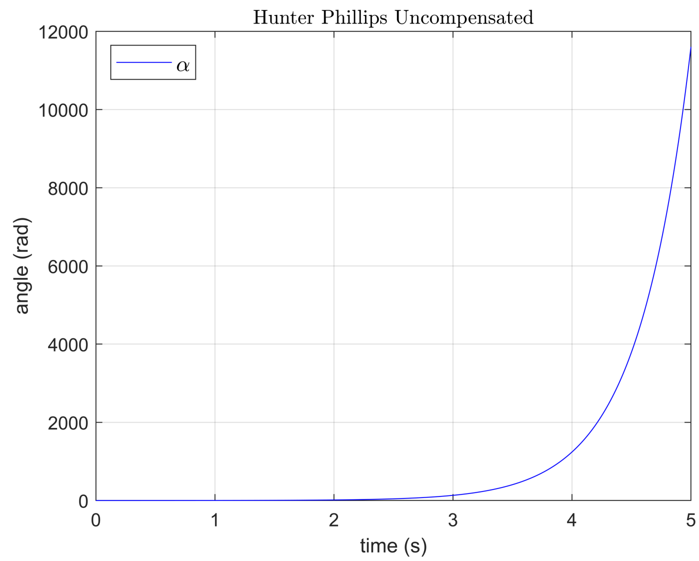

Figure 7: Plot of Angle Alpha vs Time with an Uncompensated Plant Controller

As expected, the curve balloons towards infinity with no end in sight. This provides a reference for what our controllers should be able to handle and maintain stability against.

4.2 PD Controller with No Disturbance Run

The second simulation ran was the PD controller with no disturbance as displayed in **Section 3.2**. As expected, it had an elegant approach towards the desired alpha of zero. This can be seen in *Figure 8* with a time period of 5 seconds being simulated.

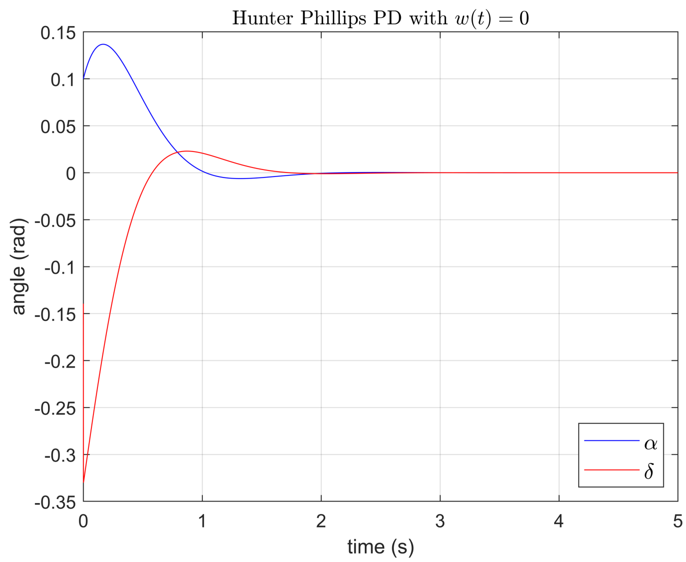

Figure 8: Plot of Angle Alpha and Delta vs Time with a PD Controller without Disturbance

This convergence is desirable for quick response systems and in an idealized test environment, this controller excels wonderfully.

4.3 PD Controller with Disturbance Run

The third simulation takes into consideration the use of a wind disturbance $w(t) = 3.5 \cdot 1(t - 4)$ being applied by a step input with the PD controller as seen in **Section 3.3**. The step was applied as seen in *Figure 9* and affected the system as shown in *Figure 10*.

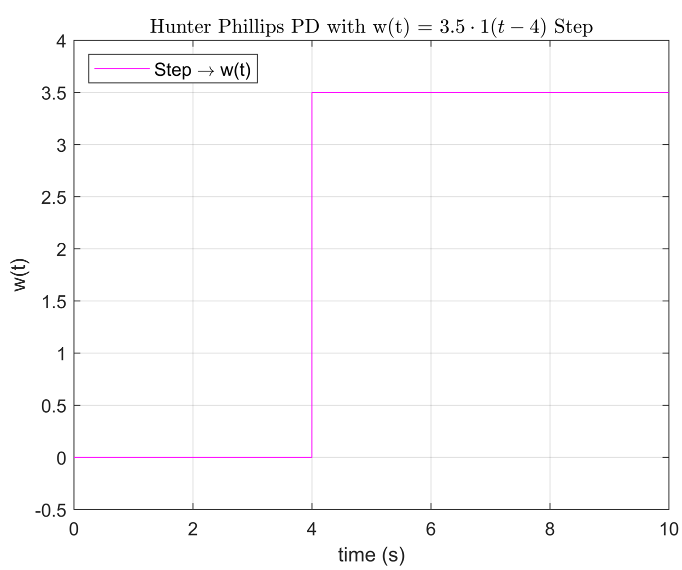

Figure 9: Plot of the Step Disturbance input w(t) for PD Controller

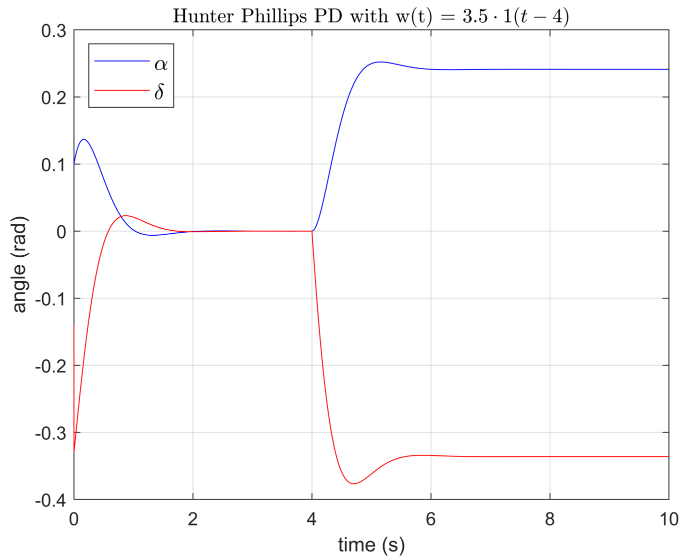

Figure 10: Plot of Angle Alpha and Delta vs Time with a PD Controller with Disturbance w(t)

It can be seen that initially the system converges nicely as seen with the controller without disturbance from in **Section 4.2**. This however is quickly affected when the step input seen in *Figure 9* is added to the system. It is observed in *Figure 10* that the system regains stability, but does not return to the desired alpha angle of zero. The controller has essentially reached its limits and has maintained the highest level of stability that it can obtain.

4.4 PID Controller with Disturbance Run

The fourth simulation uses the same step input applied in Section 4.3 being applied, however this time with a PID controller as modeled in **Section 3.4**. The wind disturbance can be seen in *Figure 11* along with the resulting effect at 4 seconds on the simulation in *Figure 12*.

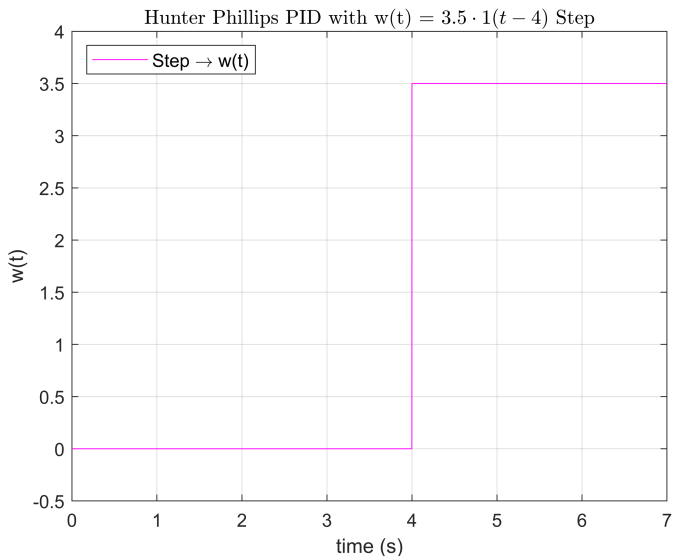

Figure 11: Plot of the Step Disturbance input w(t) for PID Controller

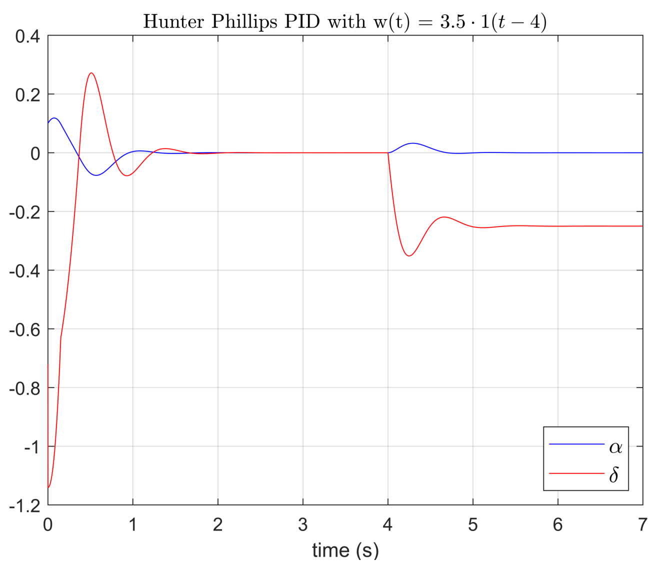

Figure 12: Plot of Angle Alpha and Delta vs Time with a PID Controller with Disturbance w(t)

As expected the PID controller handles the disturbance well and returns the alpha angle to zero fairly quickly. The delta angle is shifted with the disturbance to maintain the vertically anticipated flight path. This is all within the expected bounds and performs as desired.

5. Conclusion

Through this project, a successful rocket simulation was developed with a wide gamut of controllers. The mathematical models were derived using the problem statement to prepare for software development. A program was developed in Matlab Simulink to simulate the system and produce results for analysis. The goal being to maintain a vertical flight enabled a fluid baseline to compare each system. The uncompensated plant system did not perform well as expected. This shows why a proper controller is needed in a rocket system or the aerodynamic object would have no chance at flying vertically straight. The PD controller performed well with no external wind disturbances and had a quick response to stabilize the system. However, when a disturbance was applied, the PD controller was not able to return to an alpha of zero. It was able to stabilize to an alpha value, but not the one that the project stated as a requirement. The PID controller was able to handle this disturbance beautifully, returning alpha to zero and modifying the delta thrust angle appropriately for maintaining a stable vertical flight.

Combining all of these observations together, the following deductions can be derived. The benefit of a PD controller is that it stabilizes with a quick response time, but has the disadvantage of never returning to an alpha of zero. The advantages of a PID controller are that it stabilizes and successfully returns to zero. The drawbacks of a PID controller are that extra components are added to the system resulting in a possible source of error and extra computational time. This project concludes in a comprehensive understanding of the pitch-plane rocket stabilization problem and how to use a rocket angle-of-attack control system to simulate it effectively.

6. References

[1] Shtessel, Yuri. "Mini-Project." *Introduction to Control and Robotic Systems EE-386*, Spring 2019. The University of Alabama in Huntsville. Lectures.

[2] Dorf, R. C., & Bishop, R. H. (2016). Modern control systems (13th ed.). Boston: Pearson.