MAE 3260 Final Group Work: Exploring a System of Interest

Report

Title: It's a Bird! It's a Plane! It's a Performance Requirement of an Aircraft!

Topic of Interest: Aircraft!

Abstract: For this project, our group decided to focus on the performance requirements of an aircraft for pitch control. We chose this area because we are all very interested in aerospace, and want to apply what we've been learning in class to a more complex system. We plan to develop the model as a function of angle and, using our background from fluid dynamics, analyze how the lift and drag forces acting on the aircraft change with this angle (representing our disturbance inputs).

Students/Roles:

Student	Task/Role	Portfolio
Kat Volkova	I will define the model that we are using for the system and research/derive the system parameters to plug into future transfer functions.	https://cornell-mae-ug.github.io/fa25-portfolio-katmaxvol/projects/Aircraft%20Pitch%20Controller/
Lea Staller	I will create the state space model with the inputs we want.	
Nicholas Schneider	I will solve for the pitch equations, Laplace transform equation, and transfer function. I will also define the system's performance requirements, attempting to minimize max overshoot and response time to any inputs.	https://cornell-mae-ug.github.io/fa25-portfolio-Nicholas-K-Schneider/projects/2025-FA-MAE3260_AircraftPitchController/
Justice Rose	I will make a basic PID/LQR controller to meet the requirements using Simulink. This includes checking stability, minimizing overshoot, and ensuring good response to disturbances.	https://cornell-mae-ug.github.io/fa25-portfolio-jr2368/projects/2025-FA-MAE3260_FinalProject/

List of MAE 3260 concepts or skills used in this group work:

- Block Diagrams
- Transfer Functions
- State Space Model
- PID Controller

Citation: IEEE citation format.

Deriving System Parameters

Intro

In order to use the equations included in the provided UMichigan pitch controls document [1], we needed to define a set of parameters and constants to model our system. We chose a Boeing 747-400 model to be the specific aircraft that we are analyzing because of the breadth of data and published information about the geometry and performance coefficients of this plane.

The relevant geometric information includes the following plane characteristics [2], [4]:

Mass, m = 300000 kg
Planform area, S = 541.16 m^2
Wingspan, b = 59.63 m^2
Aspect Ratio, AR = $\frac{b^2}{S}$ = 6.57
Mean chord for 747-400, c = 8.15 m
y-y moment of inertia, i_{yy} = 33100000 slug ft^2
= 44877574.539 kg m^2

Attack angle, α = system input
Flight path angle, γ
Pitch, θ = what we are controlling using our equations
Pitch rate, q = $\frac{\theta'}{\Omega}$
Pitch rate constant, $\Omega = 2U/(mean\ chord) = (2 \times 250m/s) / 25.6m = 19.5315\ s^{-1}$

Environmental Variables

We also need some properties of the surrounding air, assuming a steady cruise altitude of 35,000 (typical for commercial aircraft). Crucially, we are assuming that the fluid surrounding the aircraft has a Reynolds number of **10^5,** which simplifies the calculations of the lift and drag coefficients. We also assume that the pressure changes over the course of the flight are so minimal that they are essentially trivial, and the downwash velocity generated by the wings is also trivial (since freestream $V_{effective} = (V_\infty^2 + V_{downwash}^2)^{½}$, and $V_\infty^2 >> V_{downwash}^2$).

Cruising altitude @ 35,000 ft, M = 0.85 or V_{cruise} = 250 m/s
Density of air @ 35,000 ft, ρ =0.371 kg/m^3
Dynamic air viscosity, μ = 1.42 x 10^{-5} Pa s

Coefficients of Performance

The final step before we can use the provided equations is to define our coefficients of performance in terms of the other variables, both geometric and environmental [2], [4].

Lift coefficient, $C_L = \frac{m*g}{\frac{1}{2}\rho V cruise^2 S} = 0.4691$.

- A big assumption that we are making is that the lift coefficient is constant. This is mostly informed by the fact that our changes in pitch are relatively minor and thus have a trivial effect, and that the freestream velocity is constant.

Drag coefficient, $C_D = C_{D0} + C_{Di} = 0.0253 + \frac{C_L^2}{\pi * AR * e}$, using Oswald efficiency factor, e = 0.7 [3].

- The drag on an aircraft is made up of 2 types: the zero-lift drag, which is a function of the geometry and velocity, and is the baseline quantity of drag at zero lift. The coefficient for this part is denoted as C_{D0}. The other component of drag is the induced drag, which is affected by the lift coefficient, the aspect ratio of the wingspan to the planform area, and an efficiency factor. The induced drag coefficient is denoted C_{Di}.

Constant $\sigma = \frac{1}{1 + \mu C_L} = \frac{1}{1 + (0.004283)(1.5)} = 0.9936.$

Weight coefficient, $C_W = \frac{W}{S \times Q} = \frac{300{,}000kg * 9.81\, m/s^2}{541.16m^2 \times (\frac{1}{2}(0.3639kg/m^3)(250m/s)^2} = 0.478$.

Moment coefficient, $C_M = \frac{Moment}{q * S * (c\ mean)}$ = zero during cruise flight, during climb ranges from -0.1 to -0.5. In our MATLAB code, we set it to equal -0.1.

Thrust coefficient, $C_T = 0.26\ s^{-1}$.

Constant $\eta = \mu\sigma C_M = -0.00013627$.

There is no publicly available information for some of the variables calculated in terms of δ, the elevator pitch angle, so we set all these calculated values equal to those shown in the U Mich web page. As Prof. Campbell stated in group work on Wednesday, these values for different types of aircraft should be very similar, so we will use the values provided in the example as an approximation for this final project.

Using the University of Michigan Pitch Control Equations

From here, we can use the following UMichigan pitch control ODEs by plugging in the performance coefficients and parameters, and run a MATLAB script to analyze motion:

$$\alpha' = \mu\Omega\sigma[-(CL + CD)\alpha + \frac{1}{(\mu - CL)}q - (CWsin(\gamma))\theta + CL)$$

$$q' = \frac{\mu\Omega}{2iyy}[[CM - \eta(CL + CD)]\alpha + [CM + \sigma CM(1 - \mu CL)]q + (\eta CWsin(\gamma)\delta]$$

$$\theta' = \Omega q$$

For this system, the input is the elevator angle δ, while the output is pitch: this gives us a pitch controller!

Plugging in the numerical values that we have, we can simplify these equations into the following:

$$\alpha' = A\alpha - Bq + C\delta$$
$$q' = -D\alpha + Eq + F\delta$$
$$\theta' = G\,q$$

We can then use MATLAB to evaluate these coefficients:

```
alpha_dot_1 =  mu * omega * sigma * (-(CL + CD)); %Solve for A

alpha_dot_2 =  mu * omega * sigma * (1 / (mu - CL)); %Solve for
B

alpha_dot_3 =  mu * omega * sigma * (-(C_w * sin(gamma) *
theta) + CL); % Solve for C

q_dot_1 = (mu * omega / (2 * I_yy)) * (C_M - eta * (CL + CD));
% Solve for D

q_dot_2 = (mu * omega / (2 * I_yy)) * (C_M + C_M * sigma * (1 -
mu * CL)); % Solve for E

q_dot_3 = (mu * omega / (2 * I_yy)) * (eta * C_w * sin(gamma));
% Solve for F

theta_dot = omega; % Solve for G
```

and get:

$$\alpha' = -0.0426\alpha - 0.1781q + 0.232\delta$$
$$q' = -0.09993\alpha - 0.1999q + 0.0203\delta$$
$$\theta' = 61.35\,q$$

Laplace Transform Equations

From the above equations, we can then define an A(S) transfer function from the α variable, a Q(S) transfer function from the q variable, a Θ(s) transfer function from the θ' variable, and a Δ (S) transfer function from the δ variable.
This gives us:

$$sA(s) = -0.0426A(s) - 0.1781Q(s) + 0.232\Delta(s)$$
$$sQ(s) = -0.09993A(s) - 0.1999Q(s) + 0.0203\Delta(s)$$
$$s\Theta(s) = 61.35Q(s)$$

Solving for Transfer Function

We can then solve this system of equations for the transfer functions between Δ(S), $\Theta(s)$, and the other variables:

$$(s + 0.426)A(s) + 0.1781Q(s) = 0.232\Delta(s)$$
$$0.09993A(s) + (s + 0.1999)Q(s) = 0.0203\Delta(s)$$

Solving for $Q(s)$ / $\Delta(s)$:

$$D(s) = s^2 + 0.2425s - 0.00928$$

Numerator:

$$Q(s) = 0.0203s - 0.02232$$

$$\frac{Q(s)}{\Delta(s)} = \frac{0.0203s - 0.02232}{s^2 + 0.2425s - 0.00928}$$

$$\Theta(s) = \frac{61.35}{s}Q(s)$$

$$\frac{\Theta(s)}{\Delta(s)} = \frac{61.35}{s}\frac{Q(s)}{\Delta(s)}$$

Laplace Transfer Function

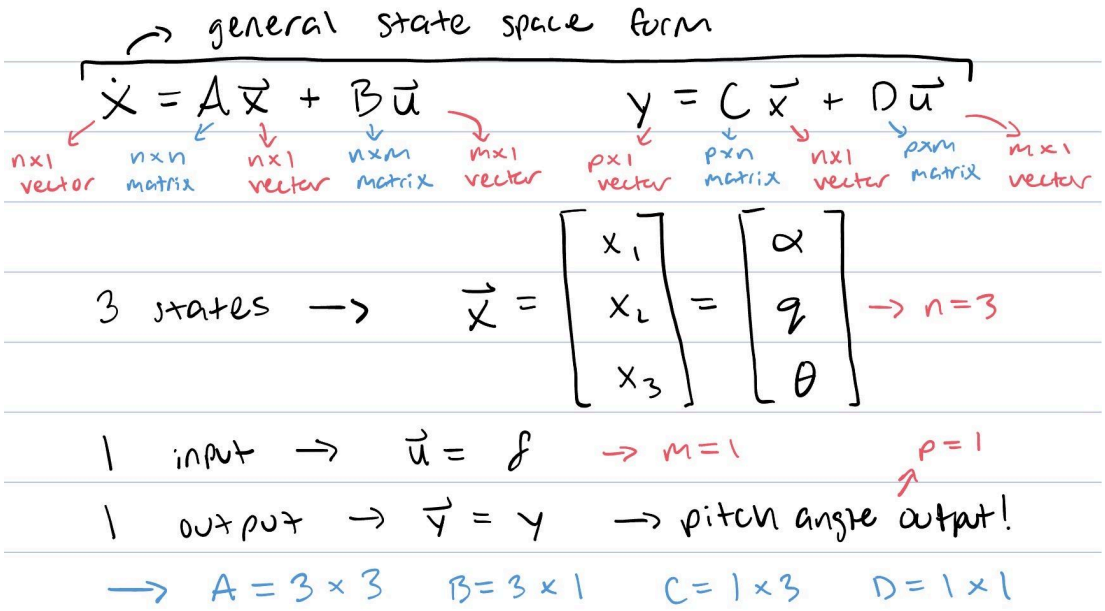
Putting all of these together gives us the transfer function:

$$P(s) = \frac{\Theta(s)}{\Delta(s)} = \frac{1.245s - 1.37}{s^3 + 0.2425s^2 - 0.00928s}$$

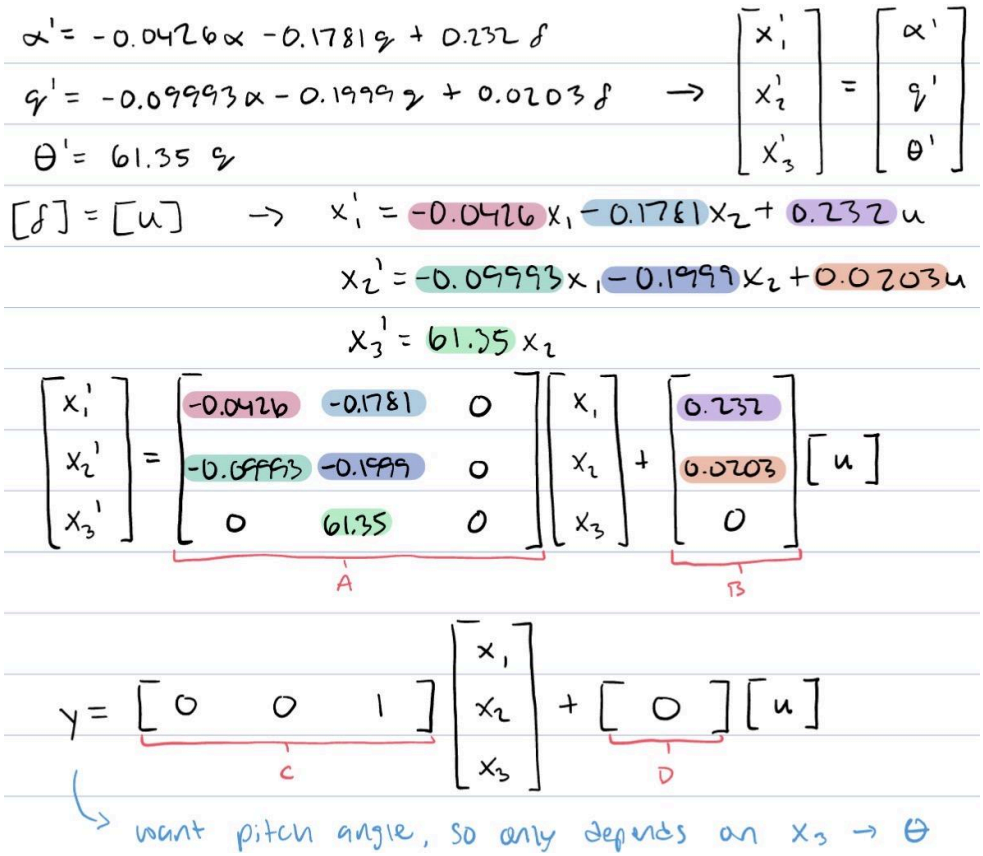
State Space

Continuing to use the modeling equations from UMichigan, we can move forward into creating the state space model for our system.

Following the methods we learned in class, we can deduce that there are three states, as there are three first order ODEs. We also know that there is one input and one output for this state space model. As outlined in the workthrough below, we can use our knowledge of how the state space is set up to find the size of the matrices, as A is a n x n, B is a n x m, C is a p x n, and D is a p x m matrix and n, m, and p are vectors based on the number of states, inputs, and outputs, respectively.



We can then move on to finding each of the entries for each matrix. Since the ODEs from UMichigan with our coefficients are in state-variable form, it made finding the entries straightforward, as we did not have to rearrange the equations. This is outlined in the photo below:



This gives us a final state space model of our system!

Controller Design Requirements

We have selected the following design criteria for our feedback controller. Our goal is to design a feedback controller that, in response to a step command input of pitch angle, overshoots by less than 10%, has a rise time of less than 2 seconds, a settling time of less than 10 seconds, and a steady-state error of less than 2%. A summary of the design requirements are below:

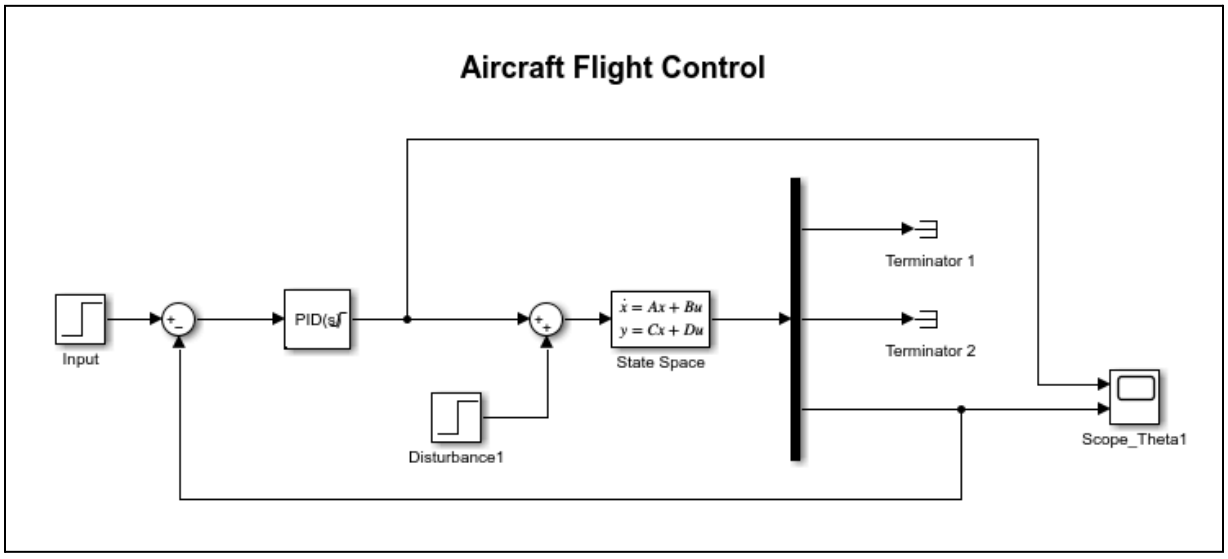
- Overshoot less than 10%
- Rise time, τ_r, less than 2 seconds
- Settling time, τ_s, less than 10 seconds
- Steady-state error, e_{ss}, less than 2%

Air Flight Controller Design - Simulink Modeling

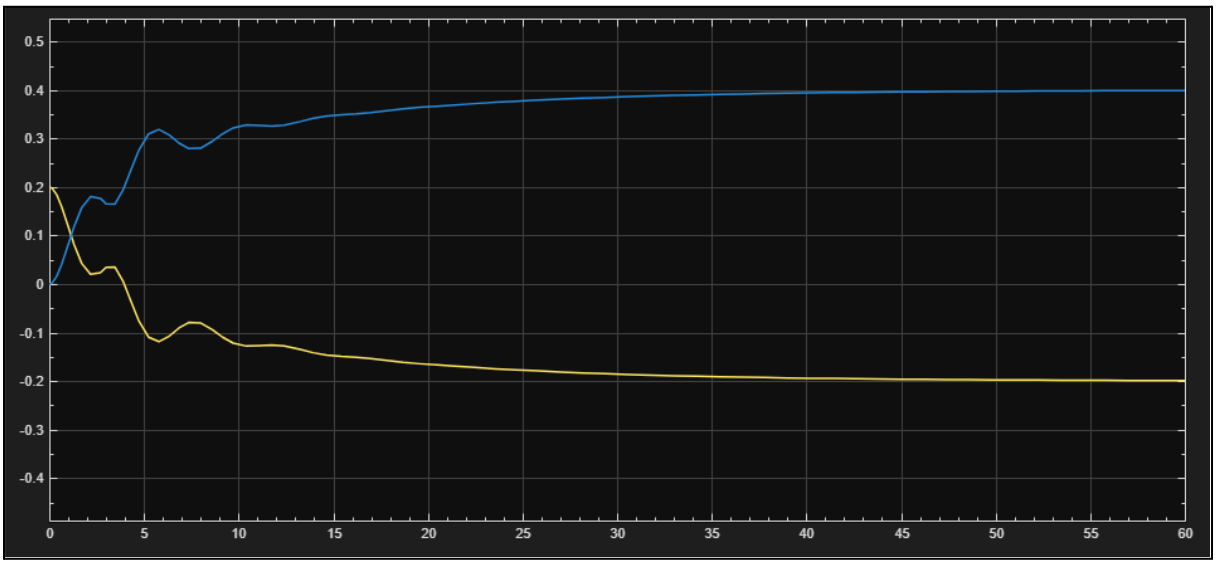
Using the α' , q' , θ' equations and state-space matrices solved above, we input this into our state-space in Simulink. As seen above, these matrices are:

$$A = [-0.0426 \ -0.1781 \ 0 ; -0.09993 \ -0.1999 \ 0; 0 \ 61.35 \ 0]$$
$$B = [0.232; 0.0203; 0]$$
$$C = [1 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 1]$$
$$D = [0; 0; 0]$$

With our state-space, we built the rest of the model around it, including an initial input, a Proportional-Integral-Derivative (PID) controller, a disturbance value of 0.2, terminators, and a scope to retrieve the results.



Prior to tuning the PID Controller, the aircraft model is operating in an open-loop configuration. In this mode, the state-space system receives the step command, but it has no feedback mechanism to correct the pitch response. As a result, the system moves purely under the influence of its internal dynamics. The blue line, representing pitch angle, has an initial increase because of the step input, but decreases with oscillation as damping increases, then levels off at an angle different from our intended goal. The yellow line, or the state derivative interaction (δ), moves with the output due to δ not being directly controlled yet.



Tuning the Controller to Meet Our Goals

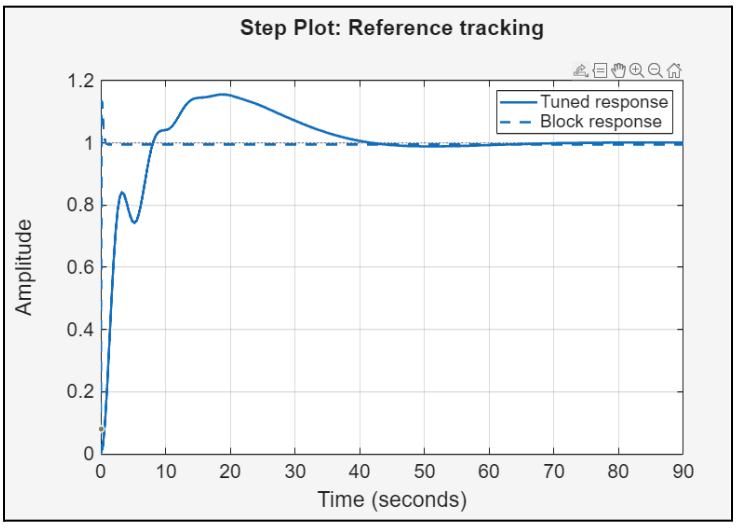
To address the deficiencies in our results, we implemented the PID controller between the reference pitch angle and the aircraft plant. The proportional term improved the responsiveness, the derivative term added damping, and the integral term eliminated steady-state error. By shifting the tuning sliders in the PID Controller toward a faster response while increasing robustness to limit overshoot, the resulting controller achieved:

Rise time ≈ 1.2 seconds

Overshoot < 10%

Settling time ≈ 7–8 seconds

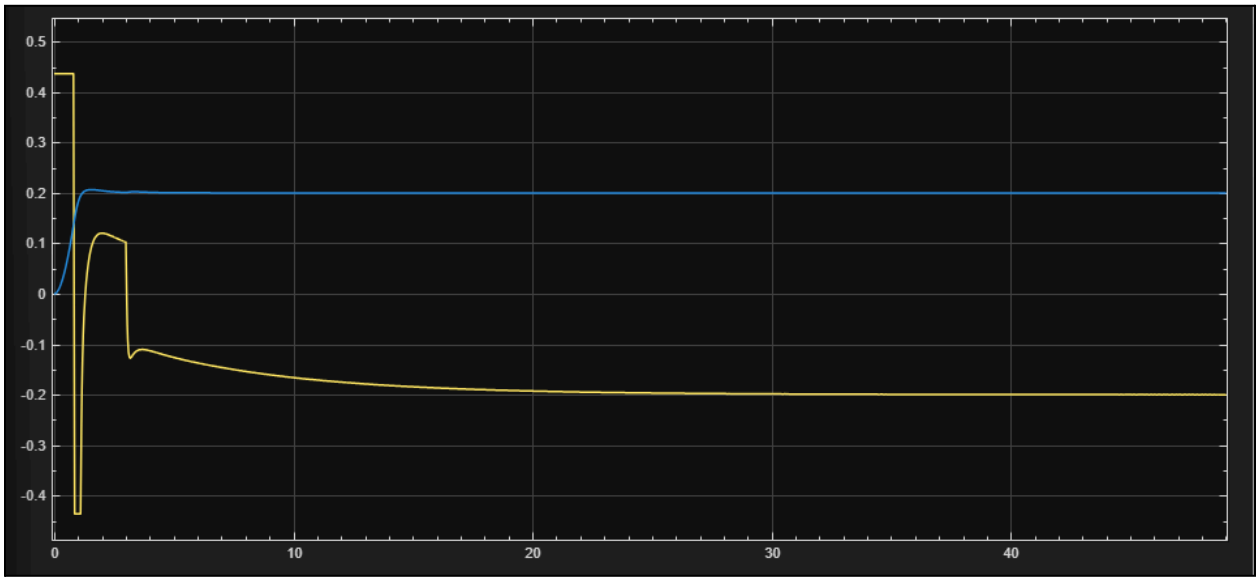
Steady-state error < 2%



The step plot shows how the now-tuned controller responds when commanded to follow a step change in pitch angle. The quick initial rise

meets the requirement for a quick rise time of less than two seconds. The low overshoot corresponds to the under 10% specification. The plot also shows the response stabilizing before 10 seconds. Finally, the steady state response very closely matches that of the ideal state.

Pitch Angle vs. Elevator Deflection Command



With the PID controller integrated into the closed-loop system, the pitch response (blue) shows a stable, well-damped trajectory. Initial disturbances are quickly suppressed, and the pitch angle settles smoothly to the commanded value with little to no steady-state error. This confirms that the PID's balance of proportional action for tracking, derivative action for overshoot control, and integral action for accuracy was successful.

The elevator deflection δ (yellow) behaves as expected for an actuator signal. It has a brief, sharp command during the initial correction phase, then is followed by smaller oscillations as the controller stabilizes the aircraft. δ converges to a steady bias, representing the aerodynamic trim required to hold the commanded pitch angle.

Overall, the PID controller achieves our project goals of stable response, minimal overshoot, quick settling, and near-zero steady-state error, while maintaining smooth behavior.

References

[1] Control Tutorials for MATLAB and Simulink, “Aircraft Pitch: System Modeling,” University of Michigan. [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=AircraftPitch&section=SystemModeling. [Accessed Dec. 4, 2025].

[2] Boeing Commercial Airplanes, *747-400 Airplane Characteristics for Airport Planning (Rev. F)*, Boeing, 2017. [Online]. Available: https://www.boeing.com/content/dam/boeing/boeingdotcom/commercial/airports/acaps/747-400_Rev_F.pdf. [Accessed Dec. 4, 2025].

[3] M. Nita and D. Scholz, “Estimating the Oswald Factor from Basic Aircraft Data,” HAW Hamburg, Report No. DLRK 12-09-10, 2012. [Online]. Available: https://www.fzt.haw-hamburg.de/pers/Scholz/OPerA/OPerA_PUB_DLRK_12-09-10.pdf. [Accessed Dec. 5, 2025].

[4] Janes All the World’s Aircraft, “Boeing 747-400,” Janes.migavia.com. [Online]. Available: https://www.janes.migavia.com/usa/boeing/boeing-747-400.html. [Accessed Dec. 4, 2025].

[5] “JSBSim Mass Properties (MassProps),” JSBSim. [Online]. Available: https://jsbsim.sourceforge.net/MassProps.html. [Accessed Dec. 5, 2025].