MAE 506: Advanced Modeling and Dynamics and Control Project

Aircraft Pitch Control

Team Number: 13

Members: Aravind Prakash Senthil, Kailash Nathan Ramahlingem, Srinivas Palaniraj, Naga Venkata Dheeraj Chilukuri, Rohit Sanjay Ganesh

1. Abstract:

This project focuses on the dynamic modeling and control of aircraft pitch motion, highlighting state-space analysis and control design techniques. The aircraft's pitch dynamics are modeled considering forces and moments, including tail lift, aerodynamic effects, and damping. A third-order system representation is developed using Laplace transforms to capture the core dynamics, integrating actuator and aerodynamic damping effects.

State-space equations are derived, with the system's controllability and observability validated through rank analysis of the respective matrices. Minimal realization and dual system properties are explored, and stability is confirmed using eigenvalue analysis, ensuring internal and BIBO stability. Controllers, including PID and LQR, are designed to meet specified performance criteria, such as minimal overshoot and fast settling times. Observer-based compensators are developed to estimate and converge on system states.

Through comparative analysis of open-loop, PID, and LQR control responses, the study emphasizes the advantages of modern control strategies in achieving optimal pitch control. Phase portraits and stability analyses provide further insights into system behavior. This comprehensive approach combines theoretical modeling with practical controller design, demonstrating robust aircraft pitch control.

2. Introduction:

Aircraft pitch control is a critical aspect of aviation dynamics, governing the angular motion about the aircraft's lateral axis. Effective pitch control ensures stability, maneuverability, and precision during flight, making it a fundamental challenge in aerospace engineering. This project explores the dynamic modeling and control of an aircraft's pitch motion, emphasizing state-space analysis and modern control design techniques.

The study begins by formulating the dynamic and state-space equations representing the pitch dynamics. Forces and moments, such as tail lift force, aerodynamic effects, and damping torques, are considered to create an accurate mathematical model. A third-order system is developed, incorporating actuator dynamics and aerodynamic damping, which play pivotal roles in the aircraft's response to control inputs.

Key properties of the system, including controllability, observability, and stability, are analyzed. The controllability and observability matrices are evaluated to confirm the feasibility of designing effective controllers and observers. Stability is assessed through eigenvalue analysis, ensuring robust performance under bounded input conditions.

To achieve desired performance characteristics, PID and Linear Quadratic Regulator (LQR) controllers are designed and compared based on their response to disturbances and reference inputs. Additionally, observer-based compensators are formulated to estimate internal states and enhance system reliability.

This project not only provides insights into the theoretical aspects of aircraft pitch control but also demonstrates practical controller and observer designs to achieve optimal performance. The findings contribute to a deeper understanding of modern control techniques applied to aviation systems, highlighting their importance in achieving precision and safety in aircraft operations.

3. Modelling:

Creating the LTI system:

$$Moment\ Equation \rightarrow \ J\ddot{\theta} = L_t \cdot d - M_a$$

$$Tail\ Lift\ Force \ \rightarrow \ \ L_t = C_t \cdot \delta_e$$

$$Pitching\ Moment \ \rightarrow \ M_a = C_m \cdot \vartheta$$

Substituting L_t and M_a

$$J\ddot{\theta} = C_t \cdot \delta_e \cdot d - C_m \cdot \vartheta$$

Aerodynamic damping introduces a torque proportional to the pitch rate $\dot{\theta}$:

$$M_d = -C_d \cdot \dot{\vartheta}$$

The moment equation becomes,

$$J\ddot{\theta} = C_t \cdot \delta_e \cdot d - C_m \cdot \vartheta - C_d \cdot \dot{\vartheta}$$

Taking Laplace transform,

$$Js^2\vartheta(s) + C_d s\vartheta(s) + C_m\vartheta(s) = C_t d\Delta_e(s)$$

By introducing the first order lag term between the pilot's input and the elevator deflection, we get the following equation:

$$(J\tau s^3 \ + Js^2 \ + c_d\tau s^2 \ + \ \ c_d s \ + \ c_m\tau s \ + \ c_m)\,\Theta(s) \ = \ c_t dU(s)$$

The transfer function between the input U(s) and output $\Theta(s)$ becomes:

$$\frac{\Theta(s)}{U(s)} = \frac{c_t d}{J\tau s^3 \ + Js^2 \ + c_d\tau s^2 \ + \ \ c_d s \ + \ c_m\tau s \ + \ c_m}$$

4. Analysis of properties:

System Representation:

The state space equation of our aircraft pitch model is given by,

$\dot{x} \ = \ Ax + Bu\ ,$

$y \ = \ Cx + Du$

The System matrices and the values substitutions are given in the appendix section

The state space formulated equation is given by

$$\mathrm{A} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \\ 0 & 0 & -2 \end{bmatrix} \ , \ B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \ , \ C = [1 \quad 0 \quad 0] \ , \ D = [0]$$

4.1 Diagonal Canonical Form:

$$A_{DCF} = \begin{bmatrix} -0.3 + 1.382i & 0 & 0 \\ 0 & -0.3 - 1.382\mathrm{i} & 0 \\ 0 & 0 & -2 \end{bmatrix}, \ B_{DCF} = \begin{bmatrix} 1.13 - 0.827i \\ 1.13 + 0.827i \\ 3.59 + 0i \end{bmatrix},$$

$$C_{DCF} = [0 \quad 0 \quad 0.5571] \ , \ D_{DCF} = [0]$$

4.2 Controllability:

The computed P is given by,

$$P = \begin{bmatrix} 0 & 0 & 6.4 \\ 0 & 6.4 & -16.64 \\ 2 & -4 & 8 \end{bmatrix},$$

$|P| \neq 0$, Therefore this system is Controllable.

4.3 Controller Canonical Form:

$$A_{CCF} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3.2 & -2.6 \end{bmatrix} \ , \ B_{CCF} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, C_{CCF} = [6.4 \quad 0 \quad 0], \ D_{CCF} = [0]$$

4.4 Observability:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \end{bmatrix},$$

$|Q| \neq 0$, Therefore this system is observable.

4.5 Observer Canonical Form:

$$A_{OCF} = {T_{OCF}}^{-1} * A * T_{OCF} = \begin{bmatrix} 0 & 0 & -4 \\ 1 & 0 & -3.2 \\ 0 & 1 & -2.6 \end{bmatrix} \ , \ B_{OCF} = {T_{OCF}}^{-1} * B = \begin{bmatrix} 6.4 \\ 0 \\ 0 \end{bmatrix},$$

$$C_{OCF} = C * T_{OCF} = [0 \quad 0 \quad 1], \ D_{OCF} = [0]$$

4.6 Minimal Realisation:

Transfer function of the SISO system is given by,

$$H(s) = \frac{32}{5s^3 + 13s^2 + 16s + 20}$$

Since, there is “no pole zero cancellation”, this system is in minimal realisation.

4.7 Duality of the SISO System

This system achieves duality since it is both controllable and observable.

$$A_{dual} = \begin{bmatrix} 0 & -2 & 0 \\ 1 & -0.6 & 3.2 \\ 0 & 3.2 & -2 \end{bmatrix} \quad , \quad B_{dual} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad , \quad C_{dual} = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix} \; , \quad D_{dual} = [0]$$

The controllability and observability matrices are given by,

$$P_{dual} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -0.6 \\ 0 & 0 & 3.2 \end{bmatrix}, Q_{dual} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \end{bmatrix}$$

4.8 Internal Stability:

The Eigen values of A are

Λ1 = -0.3+1.382i,   Λ2 = -0.3-1.382i,   Λ3 = -2+0i.

Since the real part of the eigen values are negative the system is asymptotically stable.

The computed P for the SISO system is,

$$P = \begin{bmatrix} 2.65 & 0.25 & -0.82 \\ 0 & 1.25 & 1.22 \\ -0.82 & 1.22 & 2.20 \end{bmatrix}$$

The matrix P is positive definite. Therefore, this system is asymptotically stable.

4.9 BIBO Stability:

The poles of the transfer function are given by,

$$s1 = -2 + 0i, \qquad s2 = -0.3 + 1.382i, \quad s3 = -0.3 - 1.382i$$

Since the poles of the transfer function are all having negative real parts, all the poles lie in the left half of the s plane. Therefore, this system is **BIBO Stable**.

4.10 Observer Design:

The computed L for the SISO system is,

$$L = \begin{bmatrix} 93.4 \\ 1126.96 \\ 1603.75 \end{bmatrix}$$

4.11 Observer Design:

The augmented system with both x and $\hat{x}$,

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u$$

5. Simulation Results

5.1 Observer Based Compensator:

The combined SISO system with controller and observer,

$$\begin{bmatrix}\dot{x}\\ \dot{\hat{x}}\end{bmatrix} = \begin{bmatrix}A & -BK\\ LC & A-BK-LC\end{bmatrix}\begin{bmatrix}x\\ \hat{x}\end{bmatrix} + \begin{bmatrix}BG\\ BG\end{bmatrix} r$$

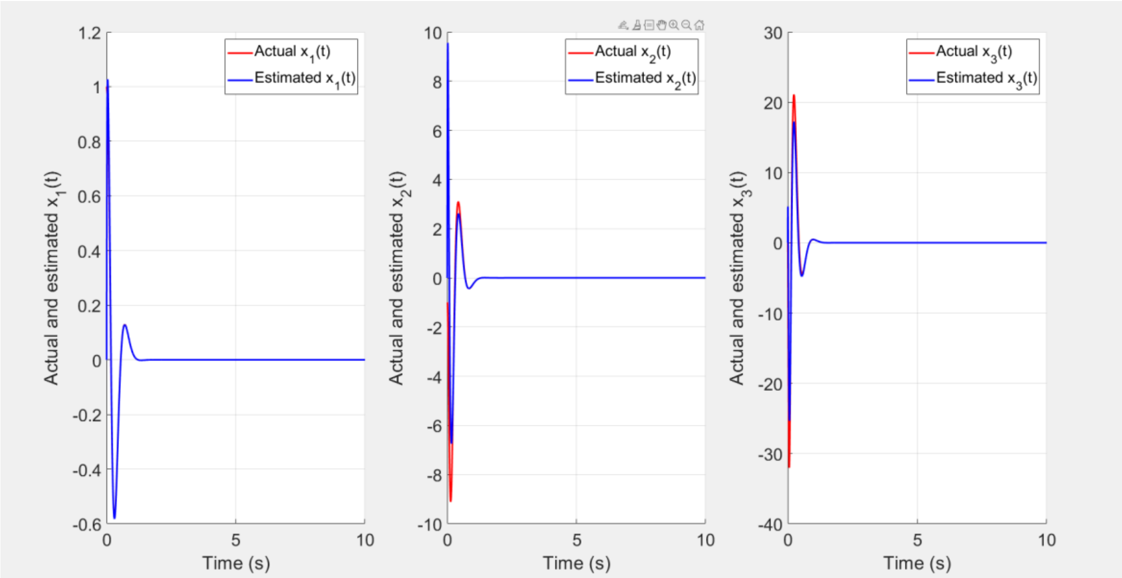


Fig 1: Actual and Obser server output plot

5.2 PID Controller Design:

Taking the required parameters: Overshoot = 1%, Settling time (t_s) = 0.5s

$$K = [1142.5 \quad 205.4 \quad 46.7]$$

Taking, $DC_{Gain} = 1$

Now constructing $\mathrm{H(s)}_{cl}$:

$$\mathrm{H(s)}_{cl} = (C^{-1}(SI - A - BK)B) * G$$

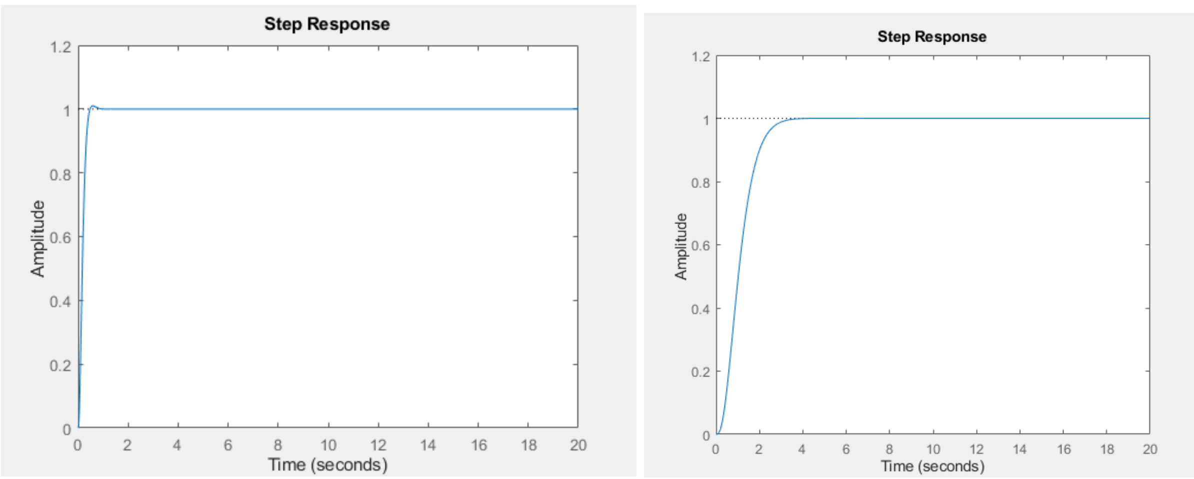


Fig 2: a) PID controller step response b) LQR controller step response

5.3 LQR Controller Design:

We have opted for a Cheap control strategy

$$K = [0.4799 \quad 2.103 \quad 2.5679]$$

5.4 Comparison between PID and LQR controller:

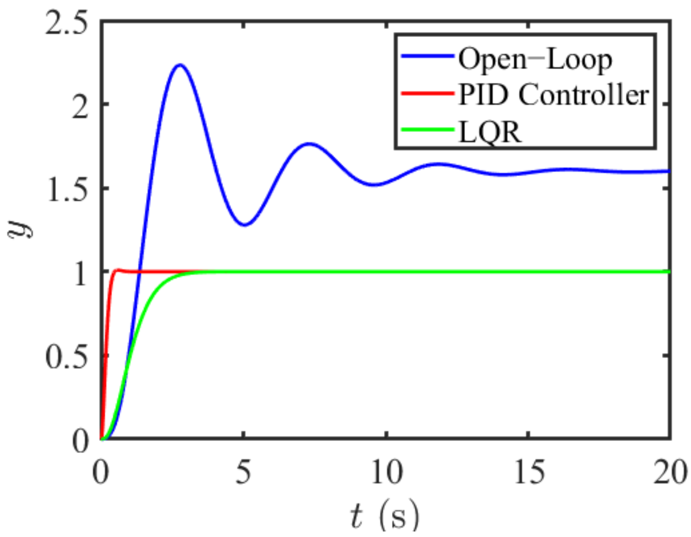


Fig 3: step response plot for the designed controllers

From the figure 3, we can see that the step response of PID controller is faster than the LQR for our set goal

Conclusion:

This project analyzed and designed control strategies for aircraft pitch dynamics, capturing core system behavior through state-space equations. Controllability, observability, and stability analyses confirmed the system's suitability for effective control and estimation.

PID and LQR controllers were designed to meet performance goals, with LQR providing an optimal "cheap" solution. Observer-based compensators accurately estimated system states, enhancing reliability. Comparative results highlighted the advantages of modern control techniques in achieving stability and precision.

Overall, the study demonstrated the importance of advanced modeling and control in ensuring safety and performance in aircraft pitch dynamics.

Reference:

1) Linear State-Space Control Systems. Robert L. Williams II and Douglas A. Lawrence Copyright 2007 John Wiley & Sons, Inc. ISBN: 978-0-471-73555-7
2) Stevens, B. L., & Lewis, F. L. (2015). *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons.
3) Cook, M. V. (2012). *Flight Dynamics Principles*. Butterworth-Heinemann.
4) NASA Technical Reports Server. (n.d.). "Stability and Control of Aircraft." Retrieved from NASA.gov

Contribution:

Aravind Prakash Senthil: 20%

Kailash Nathan Ramahlingem: 20%

Srinivas Palaniraj: 20%

Naga Venkata Dheeraj Chilukuri: 20%

Rohit Sanjay Ganesh: 20%

APPENDIX:

Creating the LTI system:

- The aircraft rotates about its centre of gravity (CG).
- Forces considered: Tail lift force L_t, weight W, aerodynamic forces.
- Moments about CG due to L_t and aerodynamic effects.

$$Moment\ Equation \rightarrow\ J\ddot{\theta} = L_t \cdot d - M_a$$

$$Tail\ Lift\ Force\ \rightarrow\ L_t = C_t \cdot \delta_e$$

$$Pitching\ Moment\ \rightarrow\ M_a = C_m \cdot \vartheta$$

Substituting L_t and M_a

$$J\ddot{\theta} = C_t \cdot \delta_e \cdot d - C_m \cdot \vartheta$$

Where,

- J: Moment of inertia about the CG.
- $\ddot{\theta}$**:** Angular acceleration (second derivative if pitch angle).
- L_t: Tail lift force proportional to elevator deflection.
- D: Distance of tail force CG.
- M_a: Aerodynamic pitching moment.

Aerodynamic damping introduces a torque proportional to the pitch rate $\dot{\theta}$:

$$M_d = -C_d \cdot \dot{\vartheta}$$

where C_d is the aerodynamic damping coefficient

The moment equation becomes,

$$J\ddot{\theta} = C_t \cdot \delta_e \cdot d - C_m \cdot \vartheta - C_d \cdot \dot{\vartheta}$$

Taking Laplace transform,

$$Js^2\vartheta(s) + C_d s\vartheta(s) + C_m\vartheta(s) = C_t d\Delta_e(s)$$

The actuator dynamics relate the input u (pilot or autopilot command) to the elevator deflection δ_r using a first-order lag:

$$\tau\dot{\delta}_e + \delta_e = u$$

Where τ is the actuator time constant.

Taking the Laplace transform:

$$\Delta_e(s) = \frac{1}{\tau s + 1}U(s)$$

Substitute this relation into the pitch equation:

$$Js^2\vartheta(s) + C_d s\vartheta(s) + C_m\vartheta(s) = C_t d\frac{1}{\tau s + 1}U(s)$$

Multiply by $(\tau s + 1)$

$$(Js^2 + C_d s + C_m)(\tau s + 1)\vartheta(s) = C_t d\frac{1}{\tau s + 1}U(s)$$

$$(J\tau s^3 + Js^2 + c_d\tau s^2 + c_d s + c_m\tau s + c_m)\,\Theta(s) = c_t dU(s)$$

The transfer function between the input U(s) and output $\Theta(s)$ becomes:

$$\frac{\Theta(s)}{U(s)} = \frac{c_t d}{J\tau s^3 + Js^2 + c_d\tau s^2 + c_d s + c_m\tau s + c_m}$$

Parameters Explained:

- J: Moment of inertia of the aircraft about the center of gravity(CG).
- C_t : Tail lift force coefficient.
- D: Distance from CG of the tail.
- C_m : Aerodynamic pitching moment coefficient.
- C_d : Aerodynamic damping coefficient.
- T: Actuator time constant.

This third- order model captures the core dynamics of pitch angle response with actuator and aerodynamic damping effects.

State Space matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-C_m}{J} & \frac{-C_d}{J} & \frac{C_t * d}{J} \\ 0 & 0 & \frac{-1}{\tau} \end{bmatrix} , \quad B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} , \quad C = [1 \quad 0 \quad 0] , \quad D = [0]$$

Here, J = 5, C_t = 8, C_m = 10, d = 2, τ = 0.5

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \\ 0 & 0 & -2 \end{bmatrix} \ , \quad B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \quad , \quad C = [1 \quad 0 \quad 0] \ , \quad D = [0]$$

Diagonal Canonical form:

Eigen values of A:

$$\lambda_1 = -0.3 + 1.382i, \qquad \lambda_2 = -0.3 - 1.382i, \qquad \lambda_3 = -2 + 0i$$

Eigenvectors associated with the above Eigenvalues:

$$T_{DCF} = \begin{bmatrix} -0.1225 - 0.5642i & -0.1225 + 0.5642i & 0.3714 \\ 0.8115 + 0i & 0.8115 + 0i & -0.7428 \\ 0 & 0 & 0.5571 \end{bmatrix},$$

$$A_{DCF} = T_{DCF}{}^{-1} * A * T_{DCF} = \begin{bmatrix} -0.3 + 1.382i & 0 & 0 \\ 0 & -0.3 - 1.382\mathrm{i} & 0 \\ 0 & 0 & -2 \end{bmatrix},$$

$$B_{DCF} = T_{DCF}{}^{-1} * B = \begin{bmatrix} 1.13 - 0.827i \\ 1.13 + 0.827i \\ 3.59 + 0i \end{bmatrix},$$

$$C_{DCF} = C * T_{DCF} = [0 \quad 0 \quad 0.5571] \quad , \quad D_{DCF} = [0]$$

Controllability:

The controllability matrix P is defined by,

$$P = [B \quad AB \quad A^2B]$$

Our P is given by,

$$P = \begin{bmatrix} 0 & 0 & \frac{C_t d}{J\tau} \\ 0 & \frac{C_t d}{J\tau} & \frac{\frac{C_t d}{J\tau} + \frac{C_t C_d d}{J^2}}{\tau} \\ \frac{1}{\tau} & \frac{-1}{\tau^2} & \frac{1}{\tau^3} \end{bmatrix}$$

The computed P is given by,

$$P = \begin{bmatrix} 0 & 0 & 6.4 \\ 0 & 6.4 & -16.64 \\ 2 & -4 & 8 \end{bmatrix},$$

|P| ≠ 0, Therefore this system is Controllable.

Controller Canonical Form:

The Characteristic Polynomial is computed as:

$$|sI - A| = \begin{vmatrix} s & -1 & 0 \\ 2 & s+0.6 & -3.2 \\ 0 & 0 & s+2 \end{vmatrix} = s^3 + 2.6s^2 + s + 4$$

Therefore, $a_0 = 4,\ a_1 = 1,\ a_2 = 2.6$

$${P_{CCF}}^{-1} = \begin{bmatrix} a1 & a2 & 1 \\ a2 & a3 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3.2 & 2.6 & 1 \\ 2.6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$T_{CCF} = P * {P_{CCF}}^{-1} = \begin{bmatrix} 6.4 & 0 & 0 \\ 0 & 6.4 & 0 \\ 4 & 1.2 & 2 \end{bmatrix},\ {T_{CCF}}^{-1} = \begin{bmatrix} 0.156 & 0 & 0 \\ 0 & 0.156 & 0 \\ -0.3 & -0.09 & 0.5 \end{bmatrix}$$

$$A_{CCF} = {T_{CCF}}^{-1} * A * T_{CCF} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3.2 & -2.6 \end{bmatrix},$$

$$B_{CCF} = {T_{CCF}}^{-1} * B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$C_{CCF} = C * T_{CCF} = [6.4 \quad 0 \quad 0],$$

$$D_{CCF} = [0]$$

Observability:

The observability matrix Q is defined by,

$$Q = \begin{bmatrix} C \\ C * A \\ C * A^2 \end{bmatrix},$$

The Observability matrix for the SISO system is given by,

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-C_m}{J} & \frac{-C_t d}{J} & \frac{C_t d}{J} \end{bmatrix},$$

The computed observability matrix is given by,

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \end{bmatrix},$$

$|Q| \neq 0$, Therefore this system is observable.

Observer Canonical Form:

$${Q_{OCF}}^{-1} = ({P_{CCF}}^{-1})^T = \begin{bmatrix} 3.2 & 2.6 & 1 \\ 2.6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$${T_{OCF}}^{-1} = {Q_{OCF}}^{-1} * Q = \begin{bmatrix} 1.2 & 2 & 3.2 \\ 2.6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$A_{OCF} = {T_{OCF}}^{-1} * A * T_{OCF} = \begin{bmatrix} 0 & 0 & -4 \\ 1 & 0 & -3.2 \\ 0 & 1 & -2.6 \end{bmatrix} \quad , \quad B_{OCF} = {T_{OCF}}^{-1} * B = \begin{bmatrix} 6.4 \\ 0 \\ 0 \end{bmatrix} ,$$

$$C_{OCF} = C * T_{OCF} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} , \quad D_{OCF} = [0]$$

Minimal Realisation:

Transfer function of the SISO system is given by,

$$H(s) = C * (sI - A)^{-1} * B + D$$

$$H(s) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \frac{5s+3}{5s^2+3s+10} & \frac{5}{5s^2+3s+10} & \frac{16}{5s^3+13s^2+16s+20} \\ \frac{-10}{5s^2+3s+10} & \frac{5s}{5s^2+3s+10} & \frac{16s}{5s^3+13s^2+16s+20} \\ 0 & 0 & s+2 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

$$H(s) = \frac{32}{5s^3+13s^2+16s+20}$$

Since, there is “no pole zero cancellation”, this system is in minimal realisation.

Duality of the SISO System

This system achieves duality since it is both controllable and observable.

The dual system is given by,

$\dot{x}_{dual} = A_{dual}x + B_{dual}u$,

$y_{dual} = C_{dual}x + D_{dual}u$,

where

$$A_{dual} = A^T, \qquad B_{dual} = B^T, \qquad C_{dual} = C^T, \qquad D_{dual} = D^T$$

$$A_{dual} = \begin{bmatrix} 0 & -2 & 0 \\ 1 & -0.6 & 3.2 \\ 0 & 3.2 & -2 \end{bmatrix} \quad , \quad B_{dual} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad , \quad C_{dual} = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix} \quad , \quad D_{dual} = [0]$$

The controllability and observability matrices are given by,

$$P_{dual} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -0.6 \\ 0 & 0 & 3.2 \end{bmatrix} , Q_{dual} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -0.6 & 3.2 \end{bmatrix}$$

Internal Stability:

The Eigen values of A are

Λ1 = -0.3+1.382i,

Λ2 = -0.3-1.382i,

Λ3 = -2+0i.

Since the real part of the eigen values are negative the system is asymptotically stable.

The Lyapunov matrix equation is given by,

$$A^T P + PA = -Q$$

The computed P for the SISO system is,

$$P = \begin{bmatrix} 2.65 & 0.25 & -0.82 \\ 0 & 1.25 & 1.22 \\ -0.82 & 1.22 & 2.20 \end{bmatrix}$$

The matrix P is positive definite since

$|P_{11}| > 0$, $\begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix} > 0$, $|P| > 0$

Therefore, this system is asymptotically stable.

Observer Design:

Desired Characteristic polynomial is given by,

$$\alpha(s) = (s - \lambda 1) * (s - \lambda 2) * (s - \lambda 3)$$

$$= (s^2 + 16s + 93.78) * (s + 80) = s^3 + 96s^2 + 1373s + 7502$$

$$\alpha_0 = 7502, \qquad \alpha_1 = 1369.8, \qquad \alpha_2 = 96$$

Observer Gain matrix is given by the Bass Gura Formula as,

$$L = \left(\begin{bmatrix} a1 & a2 & 1 \\ a2 & a3 & 0 \\ 1 & 0 & 0 \end{bmatrix} * Q \right)^T \begin{bmatrix} \alpha_0 - a_0 \\ \alpha_1 - a_1 \\ \alpha_2 - a_2 \end{bmatrix}$$

The computed L for the SISO system is,

$$L = \begin{bmatrix} 93.4 \\ 1126.96 \\ 1603.75 \end{bmatrix}$$

PID Controller Design:

Taking the parameters:

Overshoot = 1%

Settling time (t_s) = 0.5s

We can create the desired characteristic equation by finding the Natural frequency and Damping ratio.

$$\zeta_{desired} = \sqrt{\frac{\log\left(\frac{overshoot}{100}\right)^2}{\pi^2 + \log\left(\frac{overshoot}{100}\right)^2}} = 0.8261$$

$$\omega_{desired} = \frac{4}{\zeta\text{desired} * \text{Settlingtime}(t_s)} = 9.6842$$

To construct the 2nd order desired characteristic equation

$$s^2 + 2 * \zeta_{\text{desired}} * \omega_{\text{ desired}} * s + \omega_{\text{ desired}}{}^2$$

$$s^2 + 1.998 * s + 2.861$$

The roots of the 2nd order desired characteristic equation are

$$\lambda_{1,2} = -8.000115 \pm 7.717\mathrm{i}$$

The 3rd order desired characteristic equation:

$$(s^2 + 1.998 * s + 2.861) * (S + 80)$$

$$\uparrow$$

Taking the 10x the negative real parts of $\lambda_{1,2}$

Taking the coefficients of the 3rd order desired characteristic equation:

$$\alpha_0 = 7502.7$$

$$\alpha_1 = 1373.8$$

$$\alpha_2 = 96$$

$$K_{ccf} = [\alpha_0 - a_0 \quad \alpha_1 - a_1 \quad \alpha_2 - a_2]$$

$$K_{ccf} = [7498 \quad 1370.6 \quad 93.4]$$

Using the Bass Gura Formula:

$$K = K_{ccf}{}^{-1} (P * P^{-1})^{-1}$$

$$K = [1142.5 \quad 205.4 \quad 46.7]$$

To calculate the input gain (G):

$$G = \frac{-1 * DC_{Gain}}{C^{-1}(A - BK)B}$$

where, $DC_{Gain} = 1$

Now constructing $\mathrm{H(s)}_{cl}$:

$$\mathrm{H(s)}_{cl} = (C^{-1}(SI - A - BK)B) * G$$

LQR Controller Design:

We have opted for a Cheap control strategy

so the positive definite matrix (Q) is:

$$Q = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

R is a scalar that is the overall weight of the control effort

$$R = 1$$

Through the matlab coding we can obtain the K vector:

$$K = lqr(A, B, Q, R)$$

$$K = [0.4799 \quad 2.103 \quad 2.5679]$$

We can find the LQR transfer function $\mathrm{H(s)}_{lqr}$ as follows:

$$\mathrm{H(s)}_{lqr} = (C^{-1}(SI - A - BK)B) * G$$

Contents


```
clear all;
close all;
clc;

syms S
C_m =10; C_d = 3; J =5 ; C_t=8; d = 2 ; tau = 0.5;
A = [0 1 0;
     -C_m/J -C_d/J C_t*d/J;
     0 0 -1/tau]
B = [0;0;1/tau]

C = [1 0 0];
D = [0];
I = eye(3);
SI = (S*I);
disp("Find the characteristic equation for the Accf")
actual_characteristic_eqn = det(SI-A)
% Extract coefficients of the polynomial
actual_coeffs_poly = coeffs(actual_characteristic_eqn, S);

% Convert symbolic fractions to decimal
actual_coeffs_decimal = double(actual_coeffs_poly)

p_Mat = [0 0 6.4;
         0 6.4 -16.64;
         2 -4 8];
pCCFinv_Mat = [3.2 2.6 1;
                2.6 1 0;
                1 0 0];
pCCF = inv(pCCFinv_Mat);
det(pCCF);

tccf = p_Mat * pCCFinv_Mat;

tccf_inv = inv(tccf);
Accf = tccf_inv * A  * tccf;
Bccf = tccf_inv * B;
Cccf = C * tccf;

pretty(det(SI-Accf))
[V,D] = eig (A);
eigVals = diag(D);
eqn = (S-eigVals(1)) * (S-eigVals(2)) * (S-eigVals(3));




overshoot = 1;
```

```
settlingtime = 0.5;
zeta_desired = sqrt((log(overshoot/100))^2/ (pi^2 + (log(overshoot/100))^2))
omega_desired = 4 /(zeta_desired* settlingtime)


% Open-loop Response
s = tf('s');
```

Second order approximation:

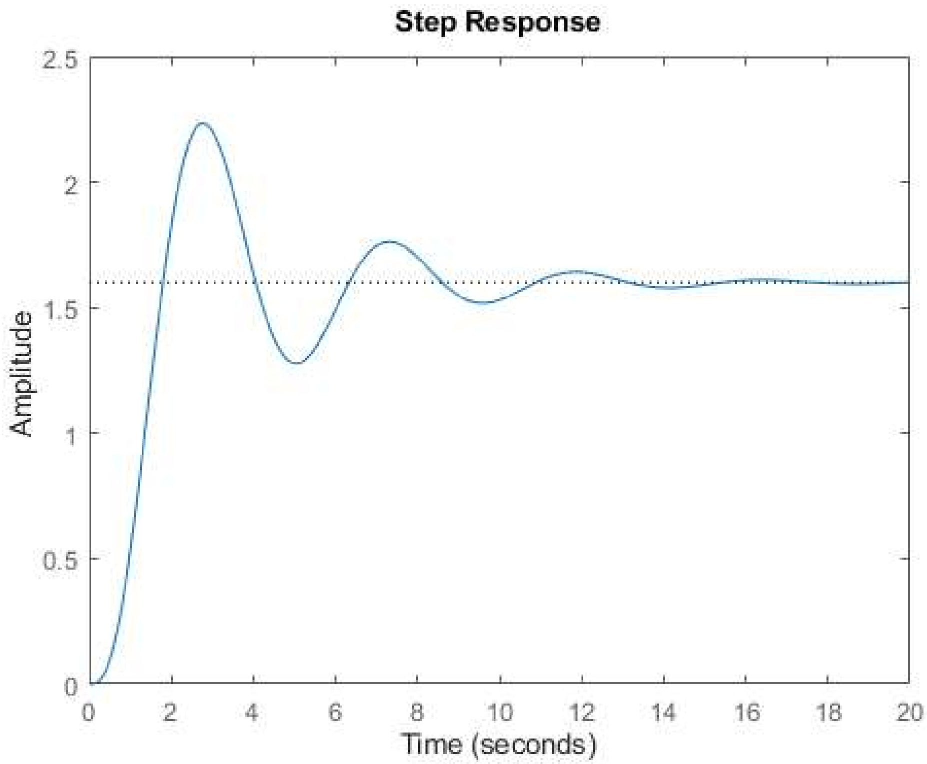
```
disp("The third order transfer function is: ")
H = (C_t*d)/((tau*s+1)*(J*s^2+C_m+C_d*s))
% Define the coefficients of the denominator polynomial
denom = [J, C_d, C_m, tau];  % Coefficients of J*s^2 + C_d*s + C_m + tau*s + 1

% Find the roots of the denominator (which are the poles)
disp("The roots of the characteristic equation is:")
poles = roots(denom)
% poles(H)We c




%2ndorder
char_eqn_desired = S^2 + (2 * zeta_desired * omega_desired * S ) + (omega_desired^2)
roots_char_eqn_desired = roots(char_eqn_desired)

%3rd order
full_char_eqn_desired = char_eqn_desired * (S + 80);

% Expand the equation to get a polynomial
expanded_eqn = expand(full_char_eqn_desired);


% Extract coefficients of the polynomial
coeffs_poly = coeffs(expanded_eqn, S);

% Convert symbolic fractions to decimal
alpha = double(coeffs_poly);
% coeffs_decimal(1)

% Find the Kccf matrix
Kccf = [alpha(1)-actual_coeffs_decimal(1) alpha(2)-actual_coeffs_decimal(2) alpha(3)-actual_coeffs_decimal(3)];

K = Kccf*inv(p_Mat*pCCFinv_Mat)
char_eqn__solution = solve(char_eqn_desired,S);


%H_OL = 1/(m*s^2+c*s+k);
H_OL = (C_t*d)/(2.5*s^3 + 6.5 *s^2 + 8*s+ C_m);
figure(1)
step(H_OL,20)
[y_OL,t1]=step(H_OL,20);


% Calculating the G value
DC_gain = 1;
G = (-1*DC_gain)/(C*inv(A-(B*K))*B)
```

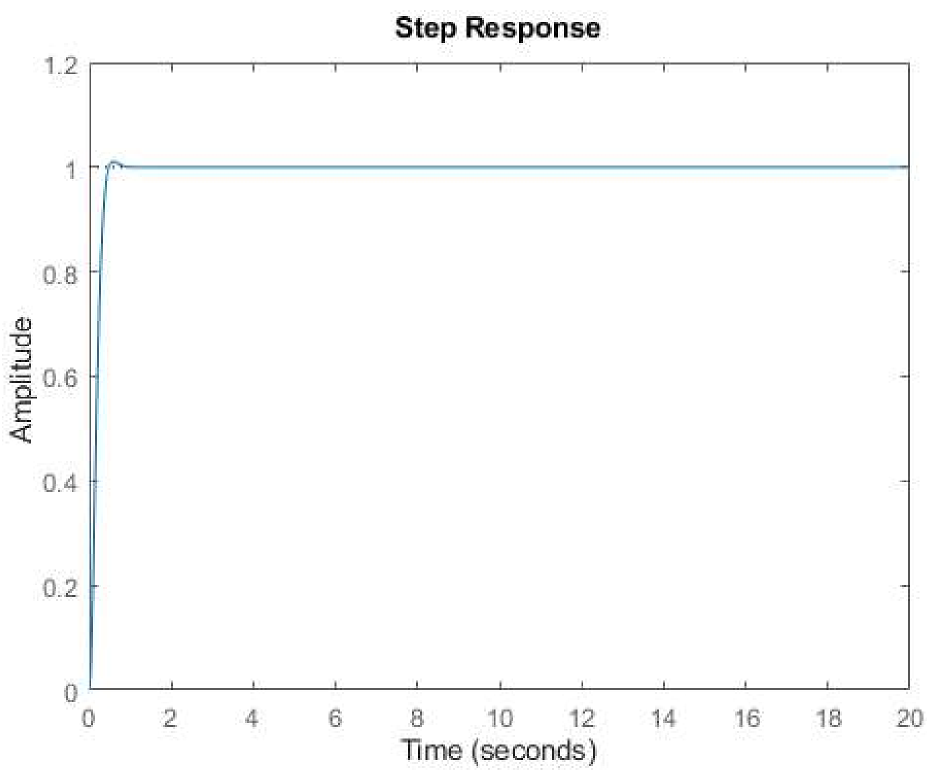
```
% Closed-loop
s = tf('s');
% H_CL = G/(2.5*s^3 + (6.5+K(2)) *s^2 + (8+(1))*s+ (10+K(3)));
% H_CL = (C-D*K_1)*inv((s.*eye(3))-(A-(B*K_1)))*B.*G + (D*G)
H_CL = ((C)*inv((s.*eye(3))-(A-(B*K)))*B)*G;
[y_CL,t2]=step(H_CL,20);
figure(2)
step(H_CL,20);



Q = eye(3) * 5;
R = 1;
K_Lqr = lqr(A,B,Q,R)
DC_gain = 1;
G_Lqr = (-1*DC_gain)/(C*inv(A-(B*K_Lqr))*B);
H_CL_LQR = ((C)*inv((s.*eye(3))-(A-(B*K_Lqr)))*B)*G_Lqr
figure(3)
step(H_CL_LQR,20)
[y_CLLQR,t3]=step(H_CL_LQR,20);


figure(4)
plot(t1,y_OL,'b','LineWidth',2)
hold
plot(t2,y_CL,'r','LineWidth',2)
plot(t3,y_CLLQR,'g','LineWidth',2)
xlabel('$t$ (s)', 'Interpreter','latex')
ylabel('$y$', 'Interpreter','latex')
legend('Open-Loop','PID Controller','LQR')
set(gca,'linewidth',2,'fontsize',20,'fontname','Times');
set(gcf,'color','white')


% Animation
% % Airplane shape (triangle)
% airplane_x = [0, -0.5, -0.5, 0]; % Airplane body x-coordinates
% airplane_y = [0, 0.25, -0.25, 0]; % Airplane body y-coordinates
% Define airplane shape
% airplane_x = [0, -0.5, -0.7, -1, -1.2, -1, -0.7, -0.5, 0]; % X-coordinates
airplane_x = [0, -0.1, -0.7, -0.9, -0.9,  0]; % X-coordinates
airplane_x = airplane_x+0.5;
airplane_y = [0,  0.2,  0.2,  0.5,    0,  0]; % Y-coordinates


% Set up figure
figure;
axis equal;
hold on;
grid on;
xlim([-2, 2]); % X-axis range for visualization
ylim([-2, 2]); % Y-axis range for visualization
xlabel('X-axis');
ylabel('Y-axis');
title('Airplane Response to Step Input');

% Plot the initial airplane
h = fill(airplane_x, airplane_y, 'b'); % Initial airplane plot
```

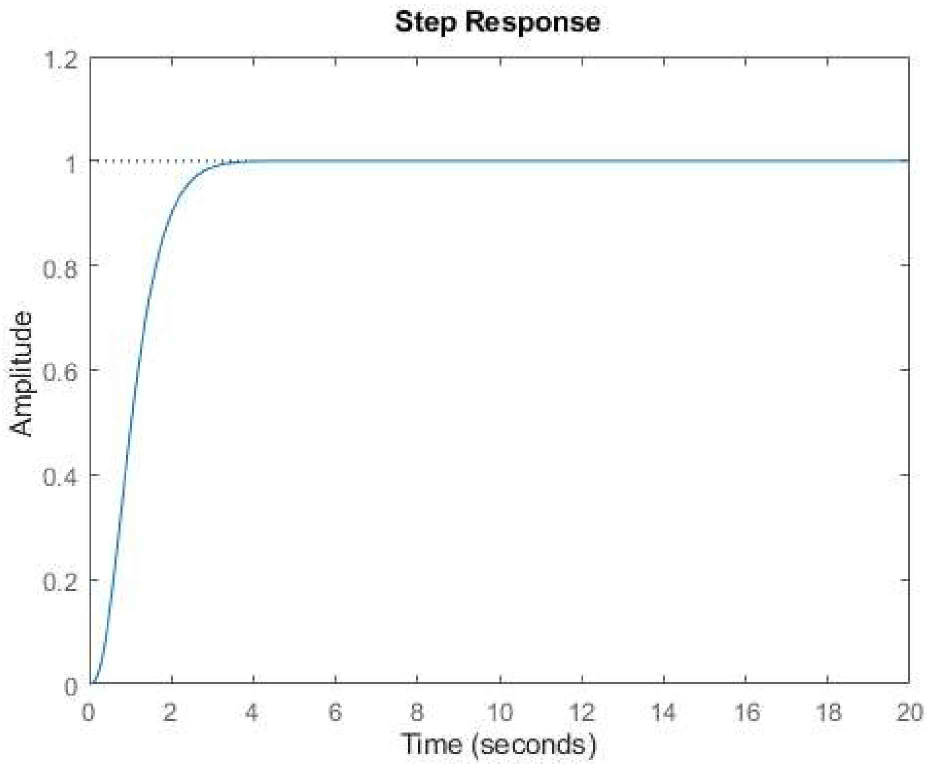
```
disp("Starting animation")
% Animate the airplane
for i = 1:200%(length(y_OL))
    % Calculate the rotation matrix for pitch angle
    theta = y_OL(i); % Pitch angle from step response (in radians)
    R = [cos(theta), -sin(theta); sin(theta), cos(theta)]; % 2D rotation matrix
    % Rotate the airplane
    rotated_coords = R * [airplane_x; airplane_y];
    set(h, 'XData', rotated_coords(1, :), 'YData', rotated_coords(2, :)); % Update airplane position

    % Pause to create animation effect
    pause(0.03);
    if i == 300
        i = length(y_OL);
    end

end
```

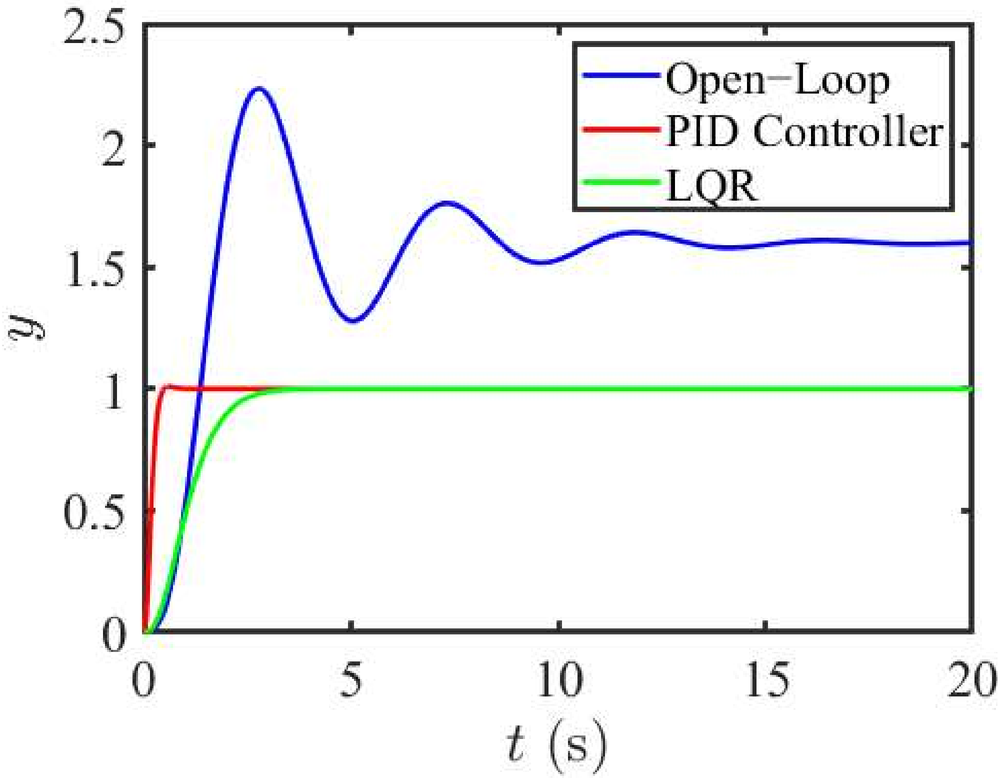
```
A =

         0    1.0000         0
   -2.0000   -0.6000    3.2000
         0         0   -2.0000


B =

     0
     0
     2

Find the characteristic equation for the Accf

actual_characteristic_eqn =

S^3 + (13*S^2)/5 + (16*S)/5 + 4


actual_coeffs_decimal =

    4.0000    3.2000    2.6000    1.0000

         2
 3   13 S    16 S
S  + ----- + ---- + 4
       5       5


zeta_desired =

    0.8261


omega_desired =

    9.6842
```

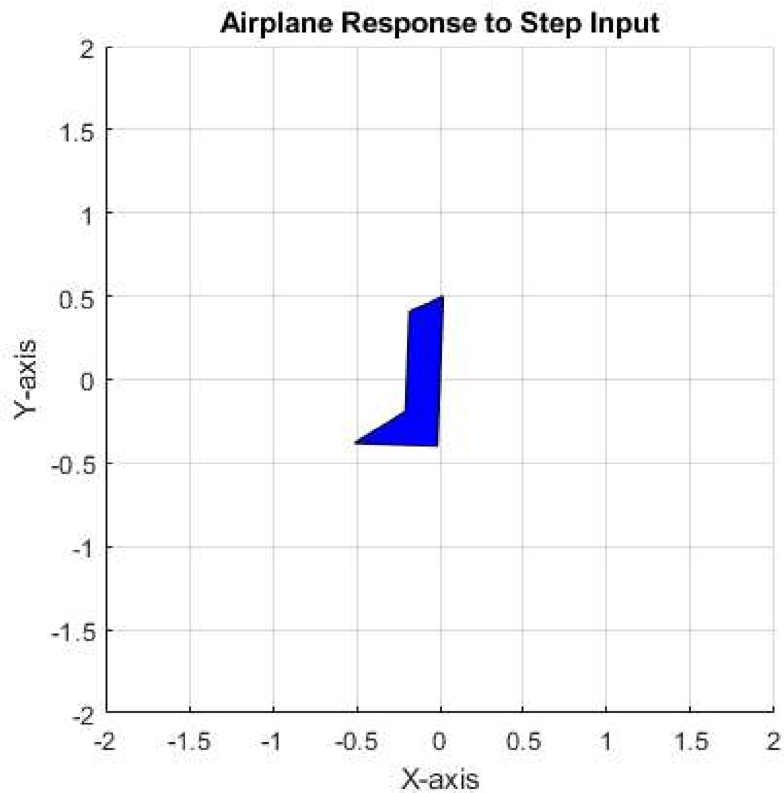
```
The third order transfer function is:

H =
 
               16
  ----------------------------
  2.5 s^3 + 6.5 s^2 + 8 s + 10
 
Continuous-time transfer function.
The roots of the characteristic equation is:

poles =

  -0.2746 + 1.3772i
  -0.2746 - 1.3772i
  -0.0507 + 0.0000i

 
char_eqn_desired =
 
S^2 + 16*S + 6599488013795445/70368744177664
 
 
roots_char_eqn_desired =
 
Empty sym: 0-by-1
 

K =

   1.0e+03 *

    1.1425    0.2054    0.0467


G =

   1.1723e+03


K_Lqr =

    0.4799    2.1033    2.5680


H_CL_LQR =
 
                17.34
  ---------------------------------
  s^3 + 7.736 s^2 + 19.74 s + 17.34
 
Continuous-time transfer function.
Current plot held
Starting animation
```


Step Response
Amplitude
Time (seconds)
0
0.5
1
1.5
2
2.5
0
2
4
6
8
10
12
14
16
18
20


Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
0
2
4
6
8
10
12
14
16
18
20


Step Response
1.2
1
0.8
0.6
0.4
0.2
0
Amplitude
0
2
4
6
8
10
12
14
16
18
20
Time (seconds)


2.5
2
1.5
1
0.5
0
y
Open−Loop
PID Controller
LQR
0
5
10
15
20
t (s)



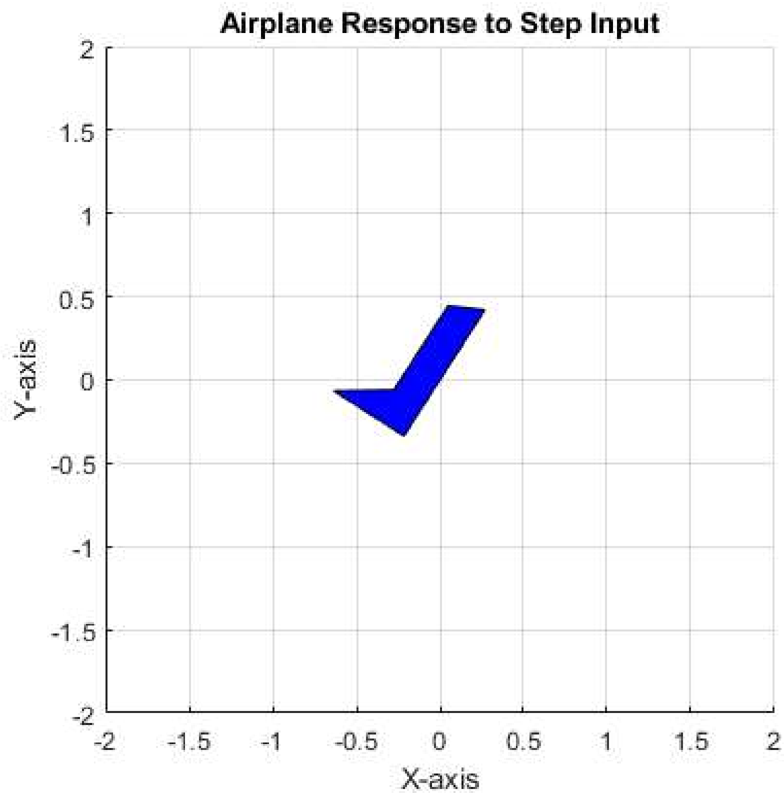
Closed loop system animation

Set up figure

```
figure;
axis equal;
hold on;
grid on;
xlim([-2, 2]); % X-axis range for visualization
ylim([-2, 2]); % Y-axis range for visualization
xlabel('X-axis');
ylabel('Y-axis');
title('Airplane Response to Step Input');

% Plot the initial airplane
h = fill(airplane_x, airplane_y, 'b'); % Initial airplane plot

disp("Starting animation")
% Animate the airplane
for i = 1:200%(length(y_CL))
    % Calculate the rotation matrix for pitch angle
    theta = y_CL(i); % Pitch angle from step response (in radians)
    R = [cos(theta), -sin(theta); sin(theta), cos(theta)]; % 2D rotation matrix

    % Rotate the airplane
    rotated_coords = R * [airplane_x; airplane_y];
    set(h, 'XData', rotated_coords(1, :), 'YData', rotated_coords(2, :)); % Update airplane position

    % Pause to create animation effect
    pause(0.03);
    if i == 300
        i = length(y_CL);
```

```
    end
end
```

```
Starting animation
```



LQR closed loop system

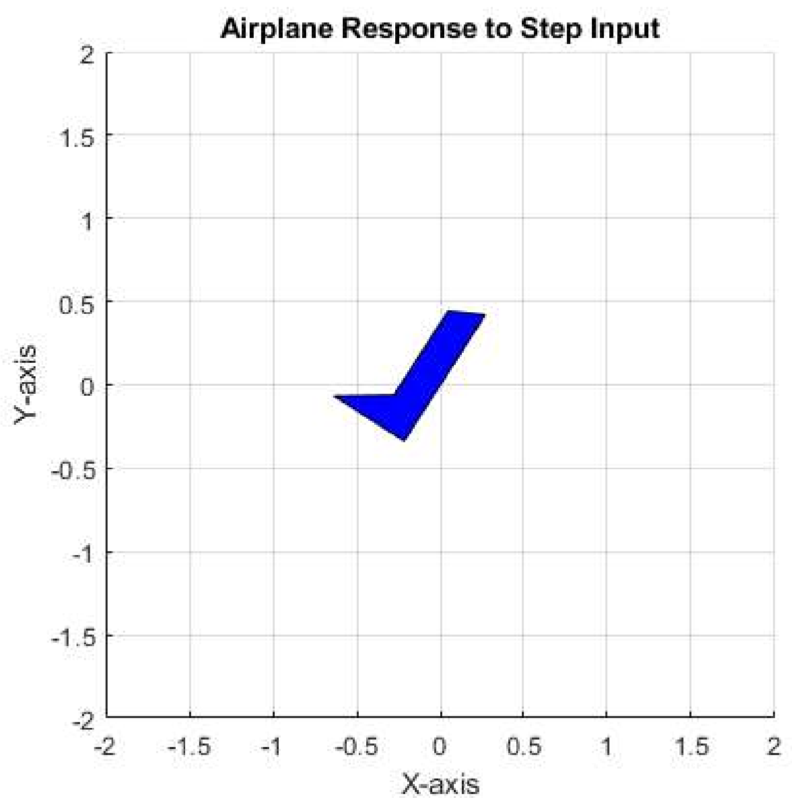
Closed loop system animation

Set up figure

```
figure;
axis equal;
hold on;
grid on;
xlim([-2, 2]); % X-axis range for visualization
ylim([-2, 2]); % Y-axis range for visualization
xlabel('X-axis');
ylabel('Y-axis');
title('Airplane Response to Step Input');

% Plot the initial airplane
h = fill(airplane_x, airplane_y, 'b'); % Initial airplane plot

disp("Starting animation")
% Animate the airplane
for i = 1:200     %length(y_CL)
    % Calculate the rotation matrix for pitch angle
    theta = y_CLLQR(i); % Pitch angle from step response (in radians)
    R = [cos(theta), -sin(theta); sin(theta), cos(theta)]; % 2D rotation matrix
```

```
    % Rotate the airplane
    rotated_coords = R * [airplane_x; airplane_y];
    set(h, 'XData', rotated_coords(1, :), 'YData', rotated_coords(2, :)); % Update airplane position

    % Pause to create animation effect
    pause(0.03);
end
```

```
Starting animation
```



Published with MATLAB® R2023b