

1. Question 1

- (1.1) The Routh array was constructed by first rewriting the characteristic equation of the closed-loop system using the substitution  $T_I = \frac{1}{\tau_I}$  to simplify the algebra. The goal was to determine the range of  $K_C$  and  $\tau_I$  values that lead to a stable system based on Routh-Hurwitz criteria.

Question 1

$G_C = K_C + T_I \frac{1}{s} = \frac{K_C s + T_I}{s}$

1.1

$$\frac{Y(s)}{R(s)} = \frac{\frac{2}{s^3 + 4s^2 + 5s + 2} \left( \frac{K_C s + T_I}{s} \right)}{1 + \frac{2}{s^3 + 4s^2 + 5s + 2} \left( \frac{K_C s + T_I}{s} \right)}$$

$$= \frac{\frac{2K_C s + 2T_I}{s^4 + 4s^3 + 5s^2 + 2s}}{1 + \frac{2K_C s + 2T_I}{s^4 + 4s^3 + 5s^2 + 2s}}$$

$$= \frac{2K_C s + 2T_I}{(s^4 + 4s^3 + 5s^2 + 2s) + 2K_C s + 2T_I}$$

$$= \frac{2K_C s + 2T_I}{s^4 + 4s^3 + 5s^2 + (2 + 2K_C)s + 2T_I}$$

$$D(s) = s^4 + 4s^3 + 5s^2 + (2 + 2K_C)s + 2T_I = 0$$

Routh Array

4	1	5	$2T_I$
3	4	$2 + 2K_C$	0
2	$b_3$	$b_1$	$b_0$
1	$c_3$	$c_1$	$c_0$
0	$d_3$	$d_1$	$d_0$

Figure 1: Step 1 of Routh Array derivation.

$$b_3 = -\frac{1}{4} \begin{vmatrix} 1 & 5 \\ 4 & 2+2K_c \end{vmatrix}$$

$$= -\frac{1}{4} [(2+2K_c) - 5(4)]$$

$$= -\frac{1}{2} - \frac{1}{2}K_c + 5$$

$$= 4.5 - \frac{1}{2}K_c$$

$$b_1 = -\frac{1}{4} (-4(2T_I))$$

$$= 2T_I$$

$$b_0 = 0$$

$$c_3 = -\frac{1}{b_3} [4b_1 - (2+2K_c)b_3]$$

$$c_1 = 0$$

$$c_0 = 0$$

$$= -\frac{4b_1}{b_3} + (2+2K_c)$$

$$= \frac{-8T_I}{4.5 - \frac{1}{2}K_c} + 2 + 2K_c$$

$$d_3 = -\frac{1}{c_3} (-b_1 c_3)$$

$$d_1 = 0$$

$$d_0 = 0$$

$$= b_1$$

$$= 2T_I$$

Figure 2: Step 2 of Routh Array derivation.

### Conditions

$$b_3: 4.5 - \frac{1}{2}K_c > 0$$

$$4.5 > \frac{1}{2}K_c$$

$$9 > K_c$$

$$d_3: 2T_I > 0 \Rightarrow T_I > 0$$

$$c_3: \frac{-8T_I}{4.5 - \frac{1}{2}K_c} + 2 + 2K_c > 0$$

$$-8T_I + 2(4.5 - \frac{1}{2}K_c) + 2K_c(4.5 - \frac{1}{2}K_c) > 0$$

$$-8T_I + 9 - K_c + 9K_c - K_c^2 > 0$$

$$-K_c^2 + 8K_c + 9 > 8T_I$$

$$-(K_c - 9)(K_c + 1) > 8T_I$$

$$\frac{(9 - K_c)(K_c + 1)}{8} > T_I$$

$\therefore$  Final conditions:

$$\textcircled{1} \quad 9 > K_c$$

$$\textcircled{2} \quad T_I > 0 \Rightarrow \frac{1}{\tau_i} > 0 \Rightarrow \tau_i > 0$$

$$\textcircled{3} \quad \frac{(9 - K_c)(K_c + 1)}{8} > T_I$$

$\rightarrow$  +ve due to  $\textcircled{1} \rightarrow \therefore K_c > -1$  to keep this +ve  
 $\uparrow$  0 due to  $\textcircled{2}$

$\tau_i > \frac{8}{(9 - K_c)(K_c + 1)}$

$$-1 < K_c < 9$$

$$T_I > 0 \approx \tau_i > 0$$

Figure 3: Final Routh Array with derived inequalities for stability.

- (1.2) Using the inequalities derived in 1.1, the feasible set was plotted to visualize combinations of  $K_C$  and  $\tau_I$  that result in closed-loop stability. The red shaded region indicates infeasible (unstable) parameter combinations.

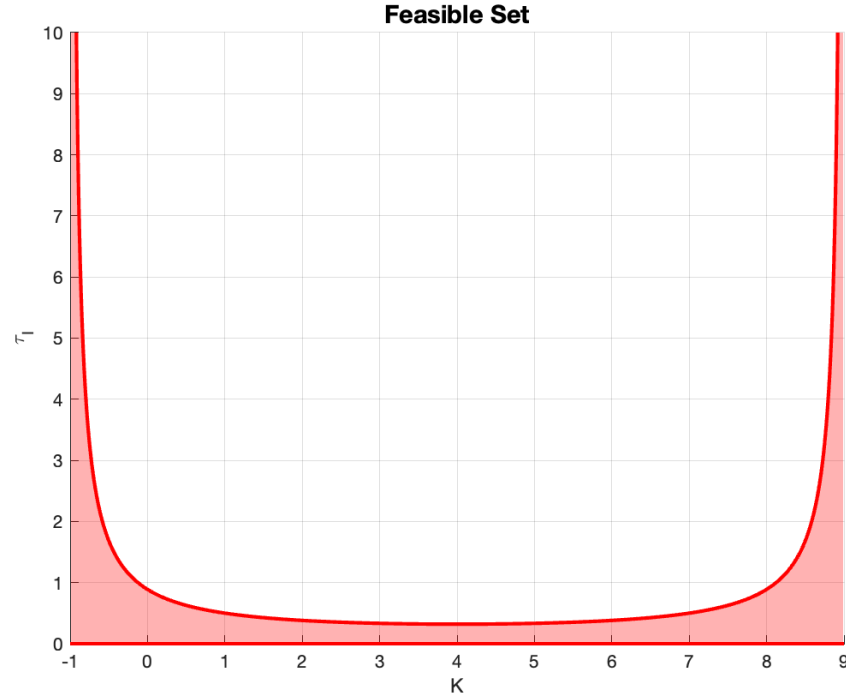


Figure 4: Feasible set for  $K_C$  and  $\tau_I$  from Routh-Hurwitz conditions.

- (1.3) A Simulink model was developed to simulate the closed-loop response of the system. Three scenarios were tested: one inside the feasible region, one outside, and one on the edge.

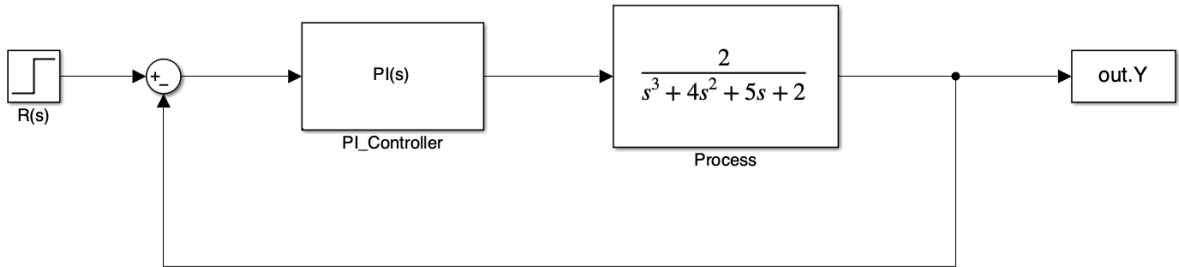


Figure 5: Simulink model of the PI-controlled system.

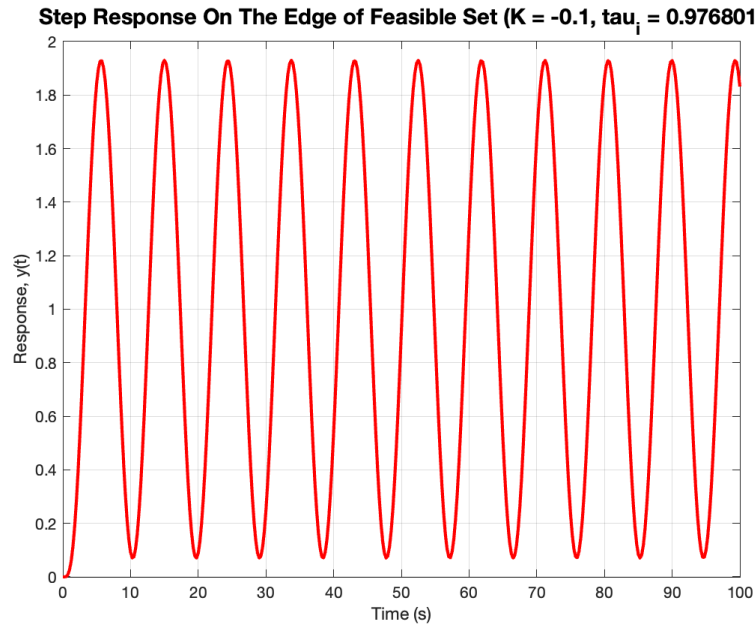


Figure 6: Step response on the boundary of the feasible set. The system oscillates and the amplitude grows very slowly, indicating marginal instability.

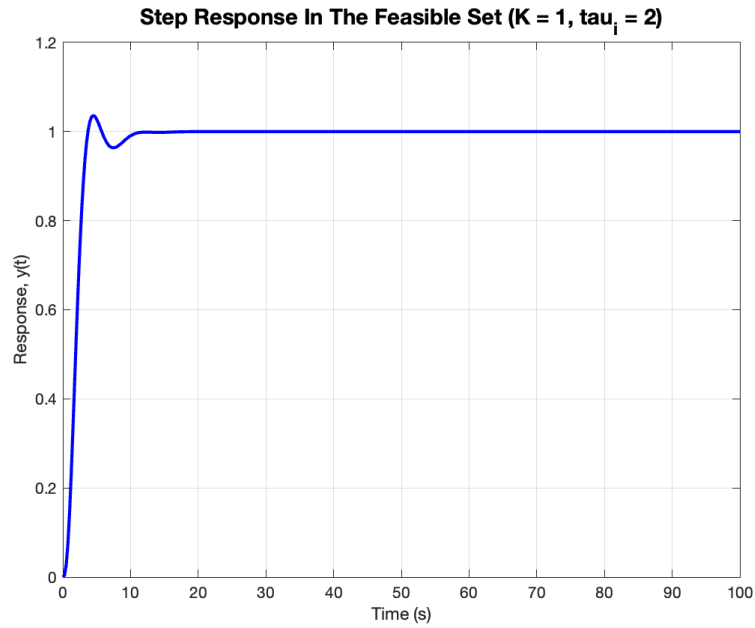


Figure 7: Step response for parameters inside the feasible set. The system is stable and relatively well damped, approaching steady state smoothly.

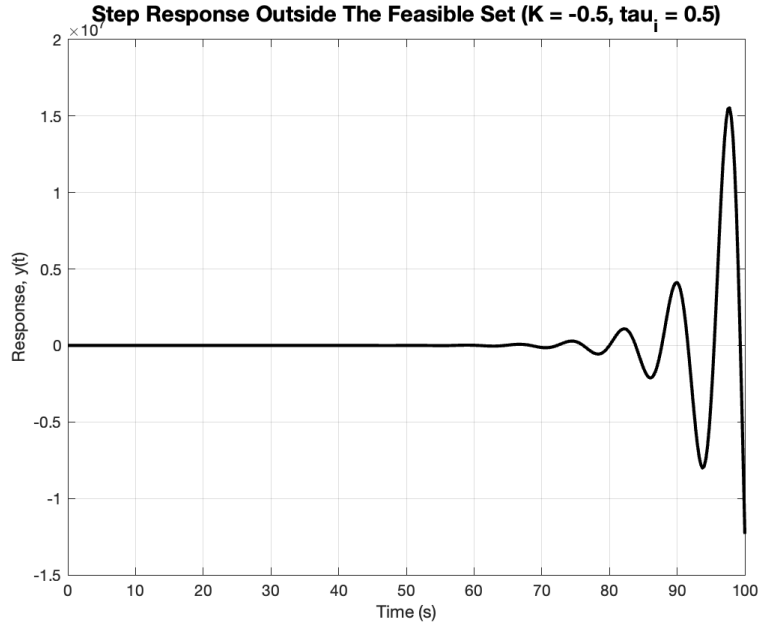


Figure 8: Step response for parameters outside the feasible set. The system exhibits fast-growing oscillations, clearly demonstrating instability.

The edge case exhibits slow-growing oscillations, which indicates marginal instability. Although the system initially appears to behave stably, the oscillations increase in amplitude over time. This is consistent with the fact that the chosen parameters lie directly on the boundary of the feasible set, where the Routh-Hurwitz conditions require strict inequalities ( $>$ ) rather than allowing equality. The second case, which is well within the feasible region, produces a stable and mostly well-damped response. The final case, with parameters outside the feasible set, immediately shows rapidly growing oscillations, confirming instability.

- (1.4) The Routh-Hurwitz array was recomputed with the new transport delay (using first-order Padé approximation) to determine how the delay affects system stability.

1.4

$$\begin{aligned}
 G_c &= K_c + T_I \frac{1}{s} = \frac{K_c s + T_I}{s} & G_H &= e^{-0.5s} \\
 & & & \approx \frac{1 - 0.25s}{1 + 0.25s} \\
 \frac{Y(s)}{R(s)} &= \frac{\frac{2}{s^3 + 4s^2 + 5s + 2} \left( \frac{K_c s + T_I}{s} \right)}{1 + \frac{2}{s^3 + 4s^2 + 5s + 2} \left( \frac{K_c s + T_I}{s} \right) \left( \frac{1 - 0.25s}{1 + 0.25s} \right)} \\
 &= \frac{(2K_c s + 2T_I)(1 + 0.25s)}{(s^4 + 4s^3 + 5s^2 + 2s)(1 + 0.25s) + (2K_c s + 2T_I)(1 - 0.25s)} \\
 &= \frac{2K_c s + 2T_I + 0.5K_c s^2 + 0.5T_I s}{s^4 + 4s^3 + 5s^2 + 2s + 0.25s^5 + s^4 + \frac{5}{4}s^3 + 0.5s^2 + 2K_c s + 2T_I - 0.5K_c s^2 - 0.5T_I s} \\
 &= \frac{0.5K_c s^2 + (2K_c + 0.5T_I)s + 2T_I}{0.25s^5 + 2s^4 + \frac{21}{4}s^3 + (5.5 - 0.5K_c)s^2 + (2 + 2K_c - 0.5T_I)s + 2T_I}
 \end{aligned}$$

$$D(s) = 0.25s^5 + 2s^4 + \frac{21}{4}s^3 + (5.5 - 0.5K_c)s^2 + (2 + 2K_c - 0.5T_I)s + 2T_I = 0$$

Routh Array

5	0.25	$\frac{21}{4}$	$2 + 2K_c - 0.5T_I$
4	2	$5.5 - 0.5K_c$	$2T_I$
3	$b_4$	$b_2$	$b_0$
2	$c_4$	$c_2$	$c_0$
1	$d_4$	$d_2$	$d_0$
0	$e_4$	$e_2$	$e_0$

Figure 9: Step 1 of Routh Array with transport delay.

$$\begin{aligned} b_4 &= -\frac{1}{2} \left[ (0.25)(5.5 - 0.5K_c) - 2\left(\frac{21}{4}\right) \right] \\ &= -\frac{11}{16} + \frac{1}{16}K_c + \frac{21}{4} \\ &= \frac{1}{16}K_c + \frac{73}{16} \end{aligned}$$

$$C_1 = -\frac{1}{b_4} [2b_2 - (5.5 - 0.5K_c)b_4]$$

$$C_2 = -\frac{1}{b_4} [-2b_4]$$

$$= \frac{-2b_2}{b_4} + 5.5 - 0.5K_c$$

$$= 2T_{\pm}$$

$$C_0 = 0$$

$$= \frac{-4K_c + 1.5T_I - 4}{\frac{1}{16}K_c + \frac{73}{16}} + 5.5 - 0.5K_c \quad \text{---} \quad (5.5 - 0.5K_c) \left( \frac{1}{16}K_c + \frac{73}{16} \right)$$

$$d_4 = -\frac{1}{c_4} [(b_4)c_2 - b_2(c_4)] \quad d_2 = 0$$

$$= b_2 - \frac{2T_I b_4}{c_4} \quad d_0 = 0$$

$$= 2K_c - 0.75T_I + 2 - \frac{(\frac{1}{16}K_c + \frac{73}{16})(2T_I)(K_c + 73)}{(-\frac{1}{2}K_c^2 - 95K_c + \frac{675}{2} + 24T_I)}$$

$$c_4 = \frac{-\frac{1}{32}K_c^2 - \frac{95}{16}K_c + \frac{675}{32} + 1.5T_I}{\frac{1}{16}K_c + \frac{73}{16}}$$

$$e_4 = -\frac{1}{d_4} [(c_4)d_2 - c_2(d_4)] \quad e_2 = 0$$

$$= \frac{-\frac{1}{2}K_c^2 - 95K_c + \frac{675}{2} + 24T_I}{K_c + 73}$$

$$\begin{aligned}
 c_4 &= -\frac{1}{d_4} [(c_4)d_2 - c_2(d_4)] & e_2 &= 0 \\
 &= c_2 - \cancel{\frac{c_4 d_2}{d_4}} & e_0 &= 0 \\
 &= 2
 \end{aligned}$$

Figure 10: Step 2 of Routh Array with transport delay.



### Conditions

$$e_4 = 2 > 0 \quad \checkmark$$

$$b_4 = \frac{1}{16} K_c + \frac{73}{16} > 0$$

$$\textcircled{1} \quad K_c > -73$$

$$c_4 = \frac{-\frac{1}{2} K_c^2 - 95 K_c + \frac{675}{2} + 24 T_I}{K_c + 73} > 0$$

$$\textcircled{2} \quad T_I > \frac{\frac{1}{2} K_c^2 + 95 K_c - \frac{675}{2}}{24}$$

$$\textcircled{3} \quad d_4 = 2 K_c - 0.75 T_I + 2 - \frac{\left(\frac{1}{16} K_c + \frac{73}{16}\right)(2 T_I)(K_c + 73)}{\left(-\frac{1}{2} K_c^2 - 95 K_c + \frac{675}{2} + 24 T_I\right)} > 0$$

↳ used matlab + symbolab to solve for  $T_I$

Figure 11: Final Routh Array with delay-based conditions.

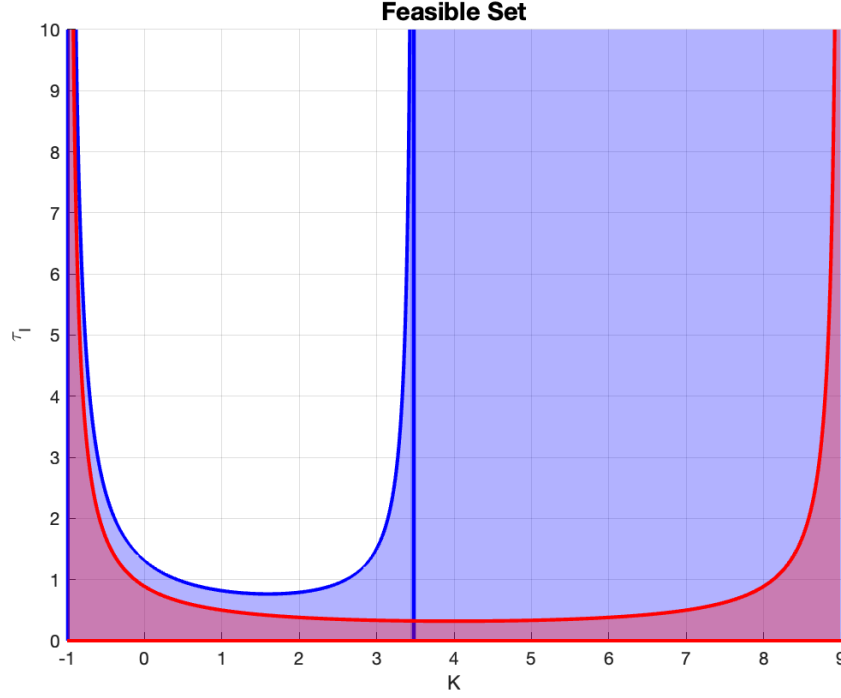


Figure 12: Feasible sets with and without delay. Blue indicates infeasible region of the new delayed system. Red is the same infeasible region from earlier.

As seen in the figure, the introduction of transport delay reduces the size of the feasible region, particularly in areas with small integral time  $\tau_I$  or large controller gain  $K_C$ . This is because transport delay introduces additional phase lag into the system, which decreases the phase margin and brings the system closer to instability. From the perspective of the Bode plot, delay shifts the phase curve to the left, meaning that the phase crosses  $-180^\circ$  at a lower frequency. This corresponds to a higher critical gain  $A_c$ , which implies a lower allowable  $K_C$  for stability. In practical terms, this means that the system must be detuned ( $K_C$  must be reduced) when delay is present. If the gain is not reduced accordingly, the aggressive controller action causes excessive oscillations, and the system approaches instability. This explains why many of the previously stable combinations of  $K_C$  and  $\tau_I$  now fall outside the feasible set when delay is introduced.

- (1.5) The system was simulated again using the same controller parameters ( $K_C = 2$ ,  $\tau_I = 1$ ) for both the delay and non-delay versions. These values were chosen by looking at the feasible sets and picking values that lay in both the time delay and non-time delayed feasible sets. The updated Simulink model includes the transport delay block.

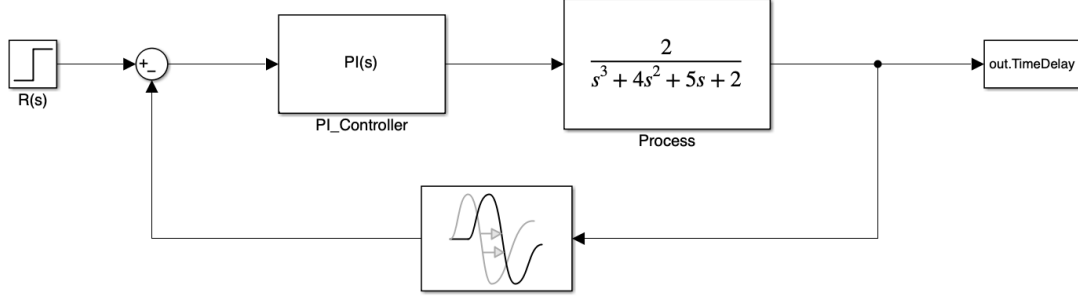


Figure 13: Simulink model with transport delay included.

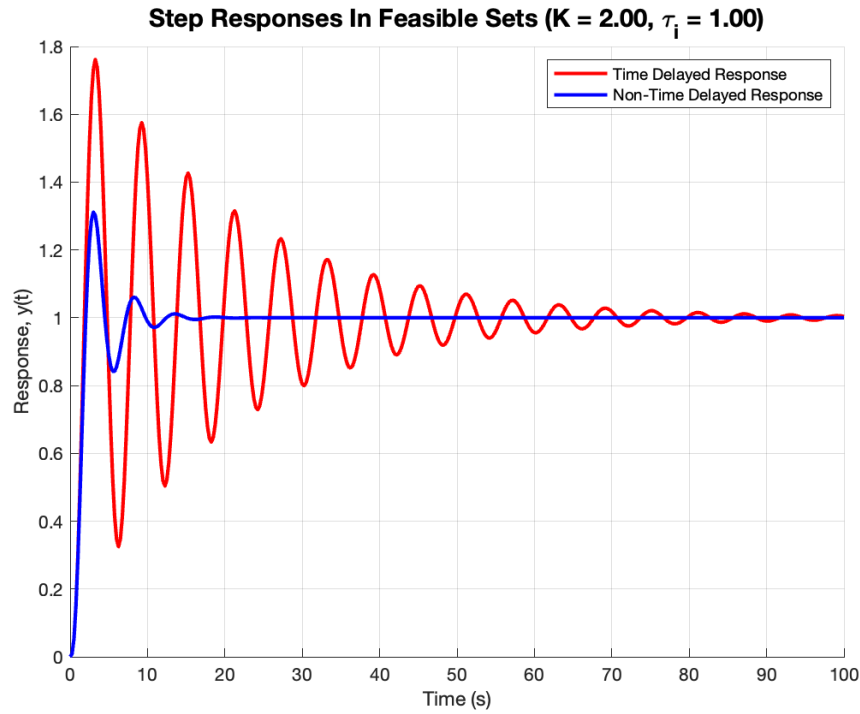


Figure 14: Comparison of system response with and without transport delay.

The system with transport delay exhibits more oscillatory behavior and takes longer to settle compared to the delay-free case, despite using identical PI tuning parameters. This aligns with the analysis in 1.4, where the added phase lag from delay was shown to reduce stability margins and narrow the feasible set. The delay effectively pushes the system closer to instability, and without detuning  $K_C$ , even stable configurations become more oscillatory and sluggish in response.

## 2. Question 2

- (2.1) The approximated first-order system equation for the closed-loop system is given by:

$$\frac{Y(s)}{U(s)} = \frac{-1.2e^{-5s}}{6s + 1} = \frac{Ke^{-\theta s}}{\tau s + 1} \quad (1)$$

We first start with the **Cohen-Coon** heuristic. From class, we are given the following equations:

$$K_C = \frac{1}{K} \cdot \frac{\tau}{\theta} \left( \frac{4}{3} + \frac{\theta}{4\tau} \right) \quad \tau_I = \theta \cdot \frac{32 + \frac{6\theta}{\tau}}{13 + \frac{8\theta}{\tau}} \quad K_D = \theta \cdot \frac{4}{11 + \frac{2\theta}{\tau}}$$

We thus get the answers below:

$$K_c = \frac{1}{(-1.2)} \left( \frac{6}{5} \right) \left( \frac{4}{3} + \frac{5}{4 \cdot 6} \right) = -1.54166$$

$$\tau_I = (5) \cdot \frac{32 + \frac{6(5)}{(6)}}{13 + \frac{8(5)}{(6)}} = 9.40677$$

$$K_D = (5) \cdot \frac{4}{11 + \frac{2(5)}{(6)}} = 1.5789$$

We then move onto the **Ciancone** heuristic. Since this system is for setpoint tracking, we use the correlation graphs as shown in class. To avoid wasting digital ink, the graphs were not attached.

For all graphs, we need to derive the fraction dead time first:

$$\frac{\theta}{\theta + \tau} = \frac{5}{5 + 6} = 0.454545$$

For the graphs  $K_c K_p$ ,  $\frac{\tau_I}{\theta + \tau}$ , and  $\frac{K_D}{\theta + \tau}$ , we get the values 0.73, 0.73, and 0.08 respectively. Solving for  $K_c$ ,  $\tau_I$ , and  $K_D$  gives us:

$$K_C = \frac{0.73}{-1.2} = -0.683 \quad \tau_I = (5 + 6) \cdot 0.73 = 8.03 \quad K_D = (5 + 6) \cdot 0.08 = 0.88$$

Lastly, we calculate the **ITAE** heuristic. We start with the equation:

$$Y = A \left( \frac{\theta}{\tau} \right)^B$$

The values of  $A$  and  $B$  are given in the table below:

Controller	Mode	A	B
PID	P	0.965	-0.85
	I	0.796	-0.1465
	D	0.308	0.929

Table 1: Controller Parameters A and B for PID Mode

We also note the following equations to calculate the parameters in set-point tracking:

$$Y_P = K K_C \quad Y_I = \frac{\tau}{\tau_I} = A + B \left( \frac{\theta}{\tau} \right) \quad Y_D = \frac{K_D}{\tau}$$

We do some mathematics:

$$Y_P = (0.965) \left( \frac{\theta}{\tau} \right)^{-0.85} = 1.12675 \quad K_C = \frac{1.12675}{-1.2} = -0.938958$$

$$Y_I = (0.796) + (-0.1465) \cdot \left( \frac{\theta}{\tau} \right) = 0.67391 \quad \tau_I = \frac{6}{0.67391} = 8.895$$

$$Y_D = (0.308) \left( \frac{\theta}{\tau} \right)^{0.929} = 0.26001 \quad K_D = 0.26001 \cdot 6 = 1.56006$$

(2.2) The Bode plot is shown below in Figure 15.

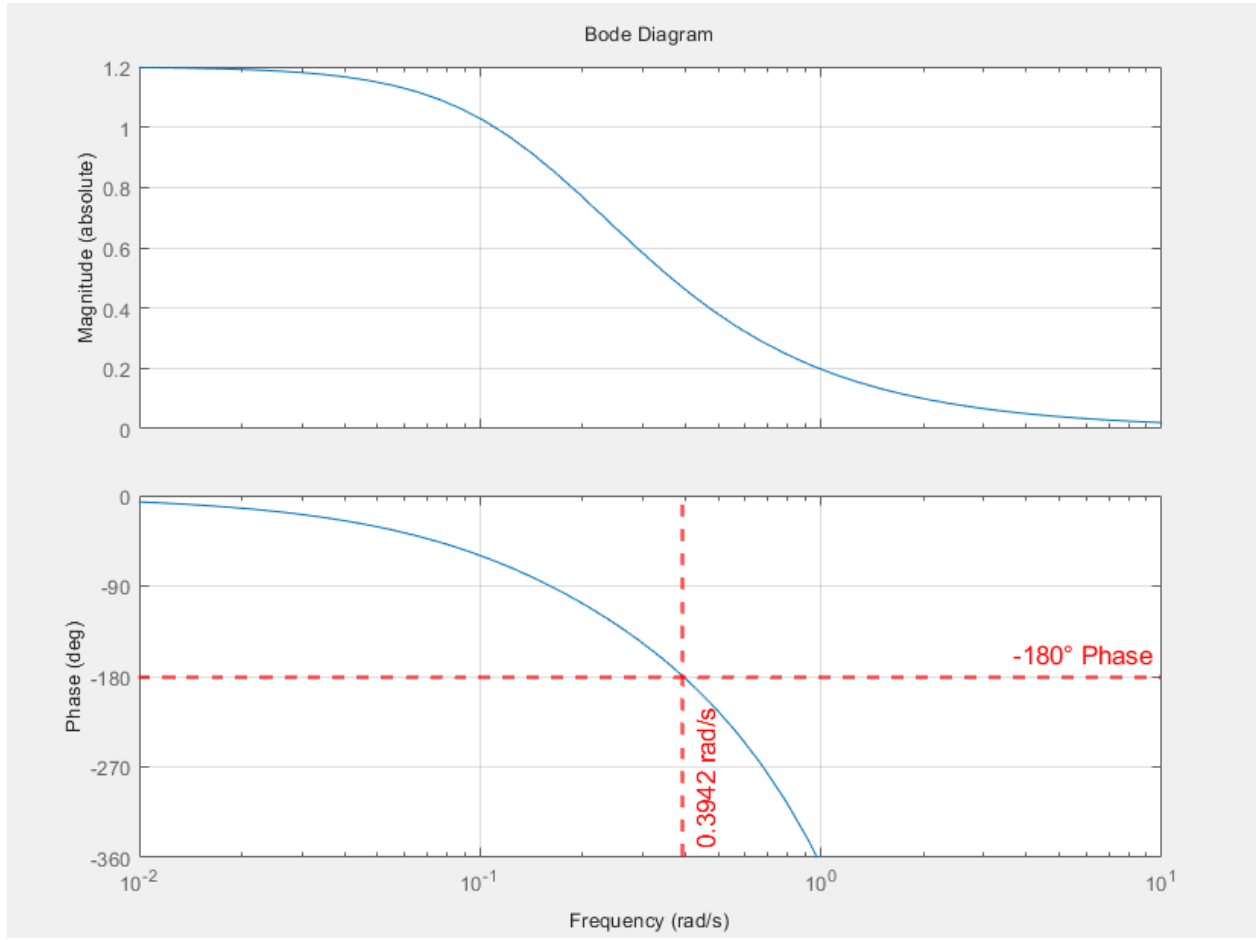


Figure 15: Bode plot of the closed-loop system with transport delay.

From the Bode plot and the associated Matlab code, we derive the Critical Frequency and the amplitude at said frequency as 0.3942 rad/s and 0.4673. With these numbers, we calculate the ultimate gain and the ultimate period.

$$K_u = \frac{1}{\text{Critical Amplitude}} = \frac{1}{0.3942} \approx 2.537$$

$$T_u = \frac{2\pi}{\omega_c} = \frac{2\pi}{0.3942} \approx 15.94$$

We then calculate the PID parameters  $K_c$ ,  $\tau_I$  and  $K_D$  using the Ziegler-Nichols tuning rules.

$$K_c = \frac{-K_u}{1.7} = \frac{-2.537}{1.7} \approx -1.492$$

$$\tau_I = \frac{T_u}{2} = \frac{15.94}{2} \approx 7.97$$

$$K_D = \frac{T_u}{8} = \frac{15.94}{8} \approx 1.992$$

- (2.3) Looking at the Figure 16 and Figure 20, the Cohen-Coon controller exhibits a more aggressive and sharp transition. Its initial decrease is steeper and reaches a larger negative value compared to the other controllers. This behavior is likely due to the high values of  $K_p$  and  $K_d$  within the controller. Additionally, the Cohen-Coon controller takes the longest time to converge, requiring close to 70 hours to stabilize.

In contrast, the Ciancone controller Figure 17 produces an output that resembles an overdamped response. While it is slower to converge compared to the ITAE and Ziegler-Nichols controllers, it is still faster than the Cohen-Coon controller, stabilizing in approximately 40 hours.

The ITAE controller (Figure 18) stands out for its faster convergence and smaller overshoot compared to the other designs. Its response is smooth and stable, making it an efficient choice.

The Ziegler-Nichols controller (Figure 19) follows a similar shape to the ITAE controller but exhibits a greater overshoot and more pronounced oscillations.

Based on the analysis of all the graphs, the ITAE controller demonstrates the best overall performance among the four designs, offering a balance of speed, stability, and minimal overshoot.

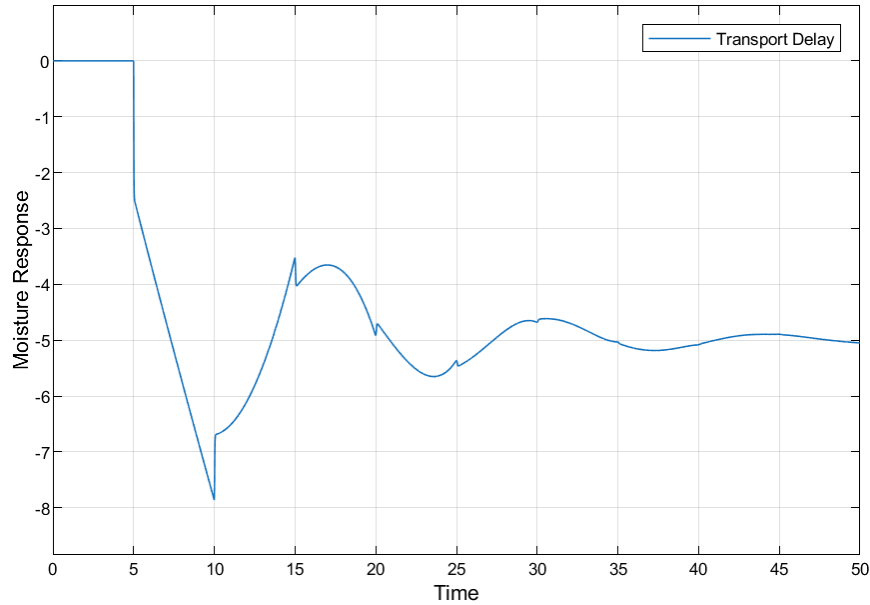


Figure 16: Step Response with Cohen-Coon Tuning

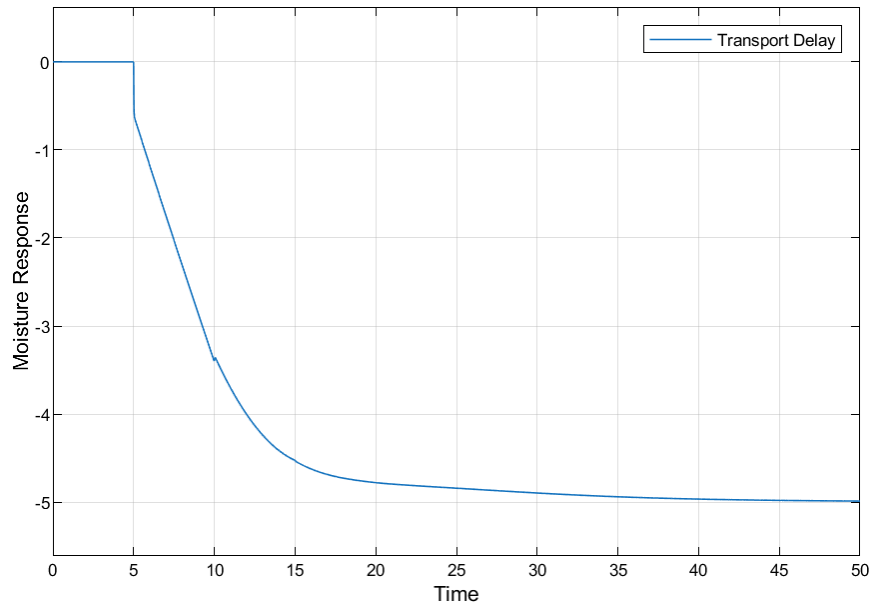


Figure 17: Step Response with Ciancone Tuning

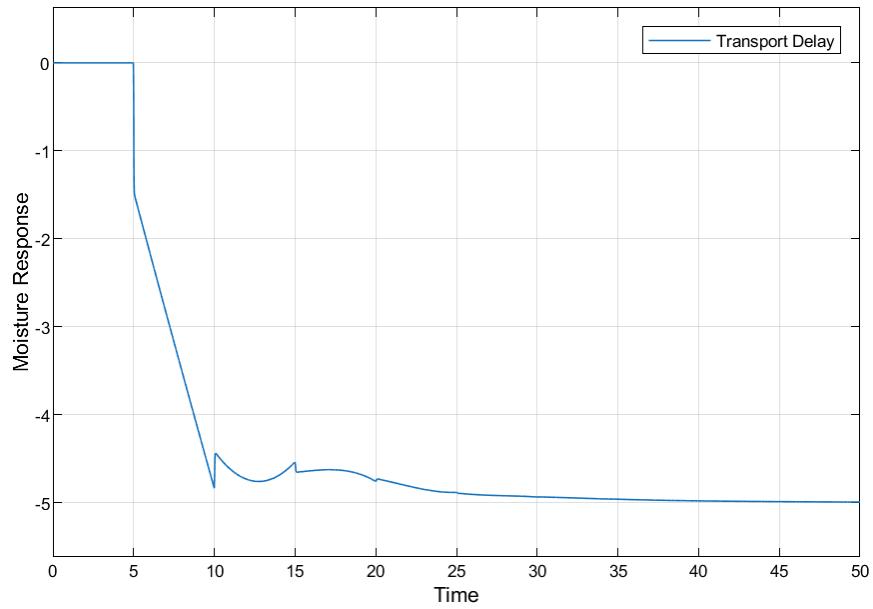


Figure 18: Step Response with ITAE Tuning



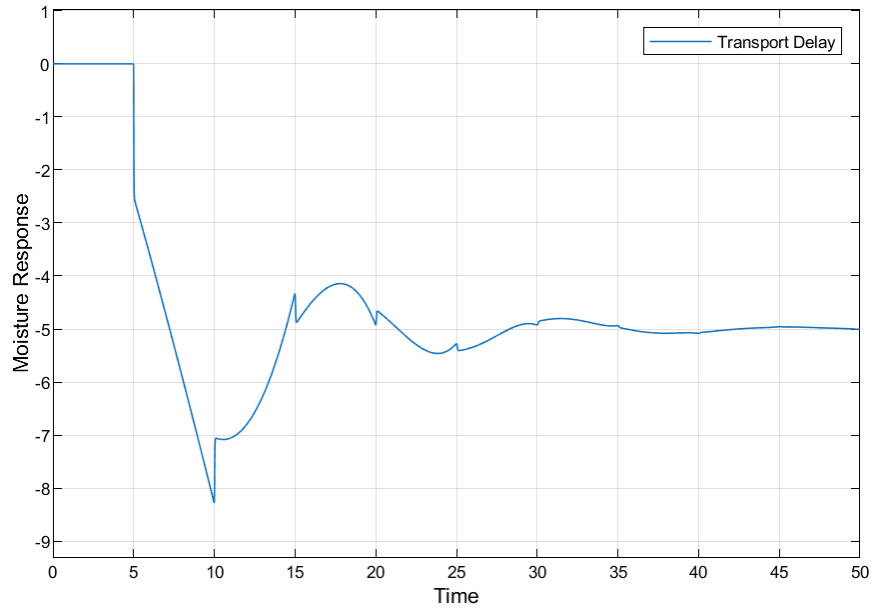


Figure 19: Step Response with Ziegler-Nichols Tuning

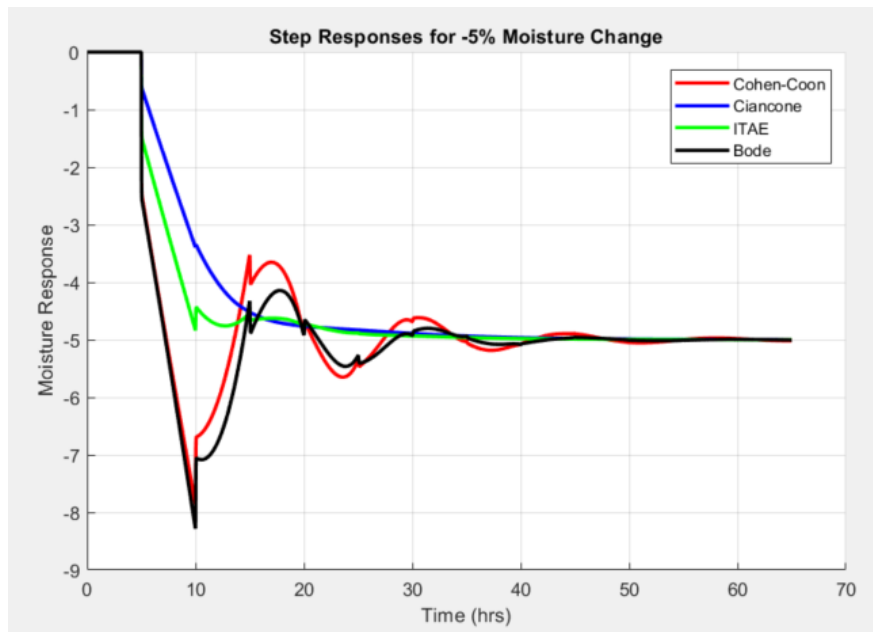


Figure 20: Comparison of Step Response with All Tuning

- (2.4) The Integral Squared Error (ISE) is a quantitative metric used to evaluate the performance of a control system. It is defined as:

$$\text{ISE} = \int_0^{\infty} e^2(t) dt$$

where  $e(t)$  is the error signal, calculated as the difference between the desired input and the system output. The ISE penalizes large errors more heavily due to

the squared term, making it useful for minimizing overshoot and oscillations in the system response.

The Integral of Time-weighted Absolute Error (ITAE) is another performance metric, defined as:

$$\text{ITAE} = \int_0^{\infty} t|e(t)| dt$$

Unlike ISE, ITAE penalizes errors that persist over time, encouraging faster convergence and smoother responses. ITAE is particularly useful for optimizing controllers in systems where stability and rapid settling are critical.

To optimize the PID controller parameters, the derivative gain ( $K_d$ ) was fixed at the ITAE value of  $K_d = 1.56006$ , while the proportional gain ( $K_c$ ) and integral time ( $\tau_I$ ) were varied. The system was simulated for a step change of  $-5\%$  moisture, and the Integral Squared Error (ISE) was calculated for each combination of  $K_c$  and  $\tau_I$ .

Figure 21 shows the contour plot of ISE values versus  $K_c$  and  $\tau_I$ . The plot reveals a clear region of optimal performance, where the ISE is minimized. The optimal parameters were identified as:

$$K_c = -1.2, \quad \tau_I = 7.5$$

These values correspond to the lowest ISE, indicating the best controller performance in terms of minimizing overshoot, oscillations, and settling time.

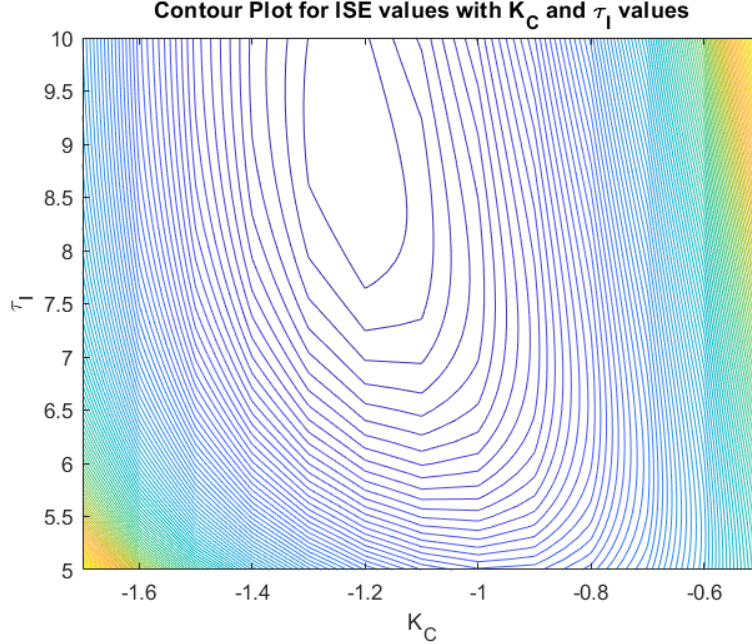


Figure 21: Contour Plot of ISE Values for Range of  $K_c$  and  $\tau_I$

The system was simulated with the optimal controller and compared to other designs, including Cohen-Coon, Ciancone, ITAE, and Bode controllers. Figure 22 shows the step responses for all controllers.

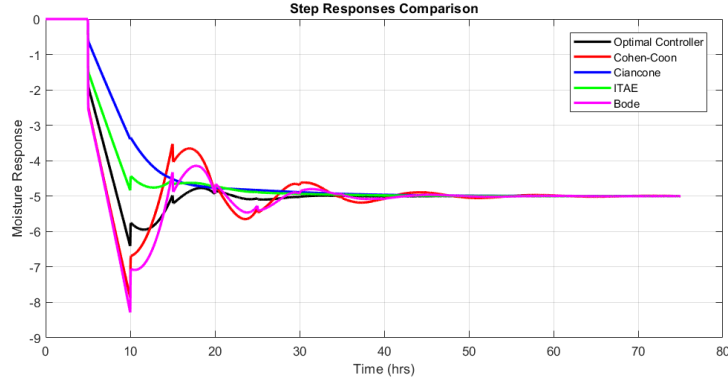


Figure 22: Contour Plot of ISE Values for Range of  $K_c$  and  $\tau_I$

The results can be summarized as follows:

- **Cohen-Coon Controller:** The Cohen-Coon controller exhibits a sharp and aggressive response, with a steep initial decrease and a larger negative overshoot. This behavior is likely due to the high values of  $K_p$  and  $K_d$ . Additionally, the Cohen-Coon controller takes the longest time to converge, stabilizing after approximately 70 hours.
- **Ciancone Controller:** The Ciancone controller produces an overdamped response, with slower convergence compared to the ITAE and Bode controllers. However, it stabilizes faster than the Cohen-Coon controller, taking approximately 40 hours to converge.
- **ITAE Controller:** The ITAE controller demonstrates smooth and stable performance, with minimal overshoot and faster convergence compared to the Cohen-Coon and Ciancone controllers. It is well-suited for systems requiring rapid settling and stability.
- **Bode Controller:** The Bode controller exhibits a similar shape to the ITAE controller but has slightly more overshoot and oscillations. While it converges quickly, it is less stable than the ITAE controller.
- **Optimal Controller:** The optimal controller, with  $K_c = -1.2$  and  $\tau_I = 7.5$ , achieves the lowest ISE and provides the smoothest response. It balances speed, stability, and minimal overshoot, making it the best choice among the designs.