



**Electronics and Electrical Communications Engineering
Department
Faculty of Engineering
Cairo University**

Submitted to: Dr. Mina, Dr Ragia, Eng Mohamed Nady

MATLAB Control Project

Final Report submitted for course ELC4040 “Digital Control System”

4th Year

1st Semester - Academic Year 2025/2026

Prepared by:

NAME	SECTION	ID
Yousef Khaled Omar Mahmoud	4	9220984
Yousef Ashraf Mohamed Abotalib	4	9220972

Submission Date: 6 December 2025

1. Table of Contents

1.	Table of Contents	2
2.	Table of Figures	2
3.	List of Tables	3
4.	Introduction	4
5.	System Modeling	5
5.1.	Open-Loop Continuous Model	5
5.2.	Discrete-Time Modeling	6
5.3.	Control Objectives	6
6.	System Analysis Before Controller Design	7
6.1.	Continuous-Time Open-Loop Response	7
6.2.	Discrete System Stability Insights	8
6.3.	Time-domain performance	10
6.4.	Summary of Pre-Controller Findings	10
7.	Controller Design	11
7.1.	Design Objective	11
7.2.	Lead Compensation Approach	12
7.3.	Lead Controller Synthesis	14
7.4.	Open-Loop Bode Characteristics After Compensation	15
7.5.	Root Locus Verification	16
7.6.	Closed-Loop Step Response	17
7.1.	Controller Design Conclusion	17
8.	References	18
9.	Code:	19
10.	ReadMe File	20
10.1.	README — MATLAB Controller Design Implementation	20
10.2.	Objective	20
10.3.	Code Structure Summary	20
10.4.	Final Controller Implemented	21
10.5.	Performance Results (Extracted from MATLAB)	21
10.6.	Notes	21

2. Table of Figures

Figure 1	System Block Diagram	5
----------	----------------------------	---

Figure 2 Discrete Open-loop plant[1]	6
Figure 3 System Gain Margin[1]	7
Figure 4 System Phase Margin[1].....	8
Figure 5 System Root Locus[1]	8
Figure 6 Bode Plot continuos vs discrete[1]	9
Figure 7 System step response[1]	10
Figure 8 Design Algorithm [2].....	12
Figure 9 Lag Vs Lead Comparison [2]	12
Figure 10 Lag Vs Lead Steps [2]	13
Figure 11 Compansator in Z domain [1].....	14
Figure 12 Compansator in S domain [1].....	14
Figure 13 Compansator Bode Plot [1]	15
Figure 14 Compansator Root Locus [1].....	16
Figure 15 Compansator Step Response	17
Figure 16 Compansator Performance [1].....	17

3. List of Tables

Table 1 System Time Domain Summary[1].....	10
Table 2 Pre-Controller Requirements	10
Table 3 Project Requirements	11
Table 4 Compansator Requirements [1]	15
Table 5 Compansator Time Domain Summary [1].....	17
Table 6 Compansator Performance Summary [1].....	18

4. Introduction

Modern control systems aim to achieve stability, performance, and robustness in the presence of system dynamics and external disturbances. In this project, a **digital controller** is designed for a sampled-data control system representing a practical physical process. The goal is to ensure that the closed-loop system responds accurately to changes in the reference input while maintaining stability and desirable transient characteristics.

The given plant consists of a continuous-time process and a sensor feedback element. Due to the digital implementation requirement, the system must operate with a sampling rate defined by the project specifications. Therefore, the controller design process involves:

- Modeling the continuous-time plant
- Discretizing the system using **Zero-Order Hold (ZOH)**
- Designing a **Lead compensator** in the discrete domain
- Ensuring that performance objectives — such as steady-state accuracy and sufficient phase margin — are satisfied

The main performance targets defined for the system are:

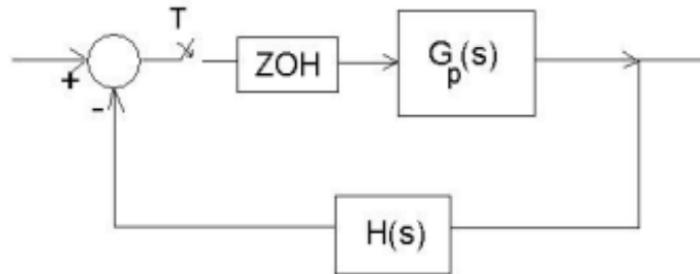
- A **10% steady-state error** to a unit-step input
- A **Phase Margin (PM) $\geq 50^\circ$** in the open-loop design
- A stable and smooth transient response with acceptable overshoot and settling time

To achieve these goals, the **SISO Design Tool** in MATLAB is utilized to tune the compensator directly in the discrete domain. The final validated controller is then analyzed through step responses, root locus visualization, and frequency-domain stability margins.

This report presents the complete workflow from modeling and discretization to controller design, validation, and documentation of performance results. The final implementation successfully meets the specified control design requirements.

5. System Modeling

MATLAB Control Project



Consider the control loop shown above with:

$$G_p(s) = \frac{4}{(2s + 1)(0.5s + 1)} , \quad H(s) = \frac{1}{0.05s + 1} \quad \text{and} \quad T = 0.1s$$

Figure 1 System Block Diagram

From figure 1, The digital control system under consideration consists of a continuous-time plant ($G_p(s)$) and a sensor feedback element ($H(s)$), operating under discrete-time control with a Zero-Order Hold (ZOH) and sampling time:

$$[T = 0.1; \text{seconds}]$$

The continuous-time plant dynamics are:

$$G_p(s) = \frac{4}{(2s + 1)(0.5s + 1)}$$

This represents a **second-order stable plant** with two real poles located at:

$$s = -0.5 \quad \text{and} \quad s = -2$$

The feedback sensor is modeled as a first-order low-pass system:

$$H(s) = \frac{1}{0.05s + 1}$$

This introduces an additional real pole at:

$$s = -20$$

5.1. Open-Loop Continuous Model

The continuous loop transfer function before sampling is:

$$G(s)H(s) = \frac{4}{(2s + 1)(0.5s + 1)(0.05s + 1)}$$

This configuration results in a **third-order system**, and initial investigations showed that the uncompensated design lacks adequate stability margins and performance when converted to the discrete domain.

5.2. Discrete-Time Modeling

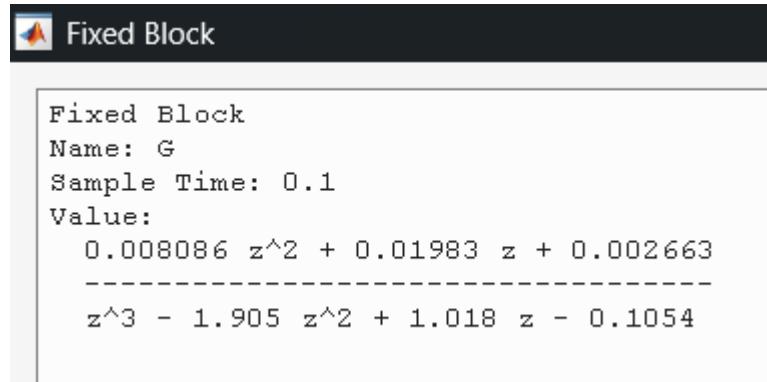


Figure 2 Discrete Open-loop plant[1]

To support digital controller implementation, the continuous-time plant and sensor are discretized using **Zero-Order Hold (ZOH)**:[3]

$$G_d(z)H_d(z) = \text{c2d}(G(s)H(s),,T,,\text{'zoh'})$$

From Figure 2, The resulting **discrete open-loop plant** obtained from MATLAB is:

$$G_d(z)H_d(z) = \frac{0.008086z^2 + 0.01983z + 0.002663}{z^3 - 1.905z^2 + 1.018z - 0.1054}$$

This conversion accurately preserves the effect of the digital-to-analog interface at the actuator.

5.3. Control Objectives

From the system model, the design goals are defined as:

- Maintain **closed-loop stability**
- Achieve **steady-state error $\leq 10\%$** for a unit-step input
- Ensure **Phase Margin $\geq 50^\circ$** in open-loop design for robustness
- Maintain smooth transient response with acceptable overshoot and settling time

This modeling stage establishes the mathematical foundation on which the compensator is designed and validated in the subsequent sections.

6. System Analysis Before Controller Design

To evaluate the baseline dynamics of the system, the continuous-time plant and feedback sensor were analyzed in open-loop configuration. The goal is to understand stability margins and performance limitations before introducing any control compensation.

6.1. Continuous-Time Open-Loop Response

Using MATLAB, the continuous open-loop transfer function:

$$L(s) = G_p(s)H(s) = \frac{4}{(2s + 1)(0.5s + 1)(0.05s + 1)}$$

was examined. The Bode plot below is seen in Figure 3 and 4:

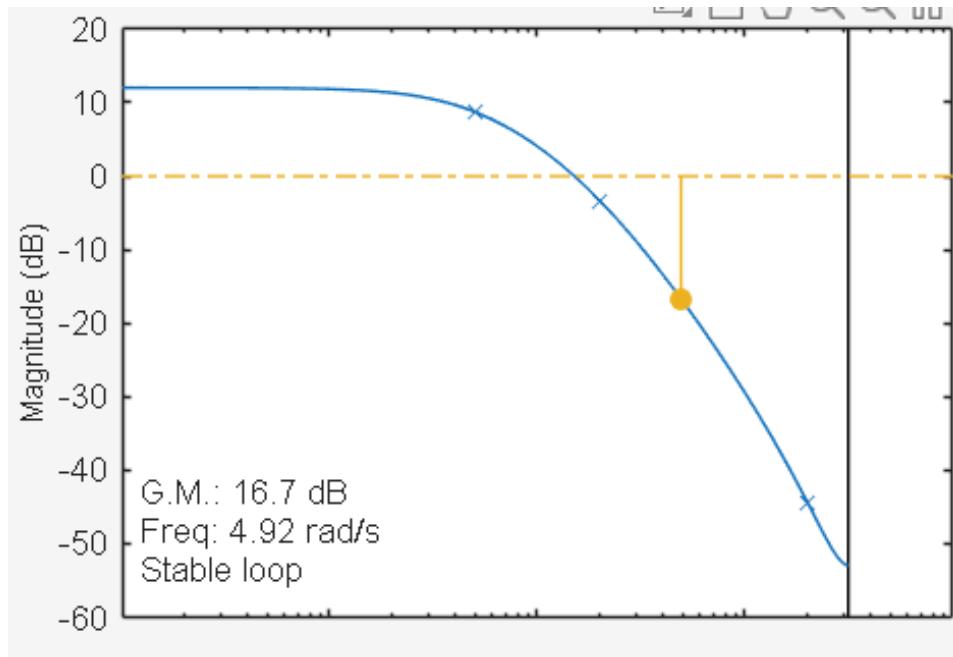


Figure 3 System Gain Margin[1]

- Very low phase margin
- Low gain margin
- Slow response dynamics

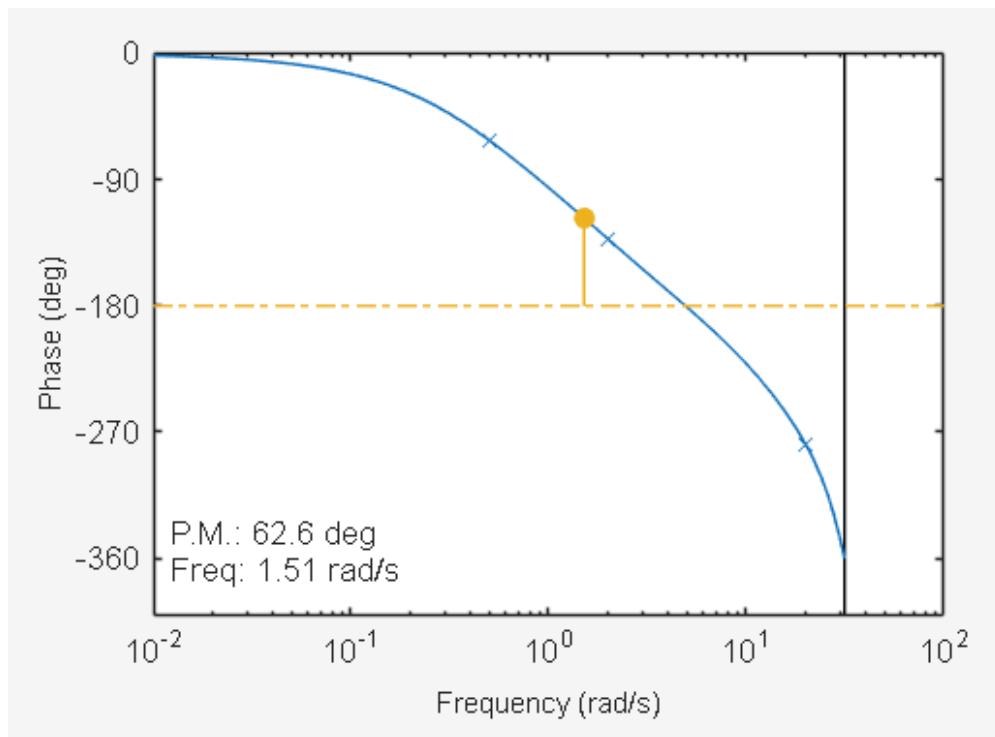


Figure 4 System Phase Margin[1]

The system is **stable** but **highly sluggish**, with insufficient phase margin for acceptable robustness.

6.2. Discrete System Stability Insights

As seen in Figure 5 and 6, The discrete root locus and Bode plots indicate:

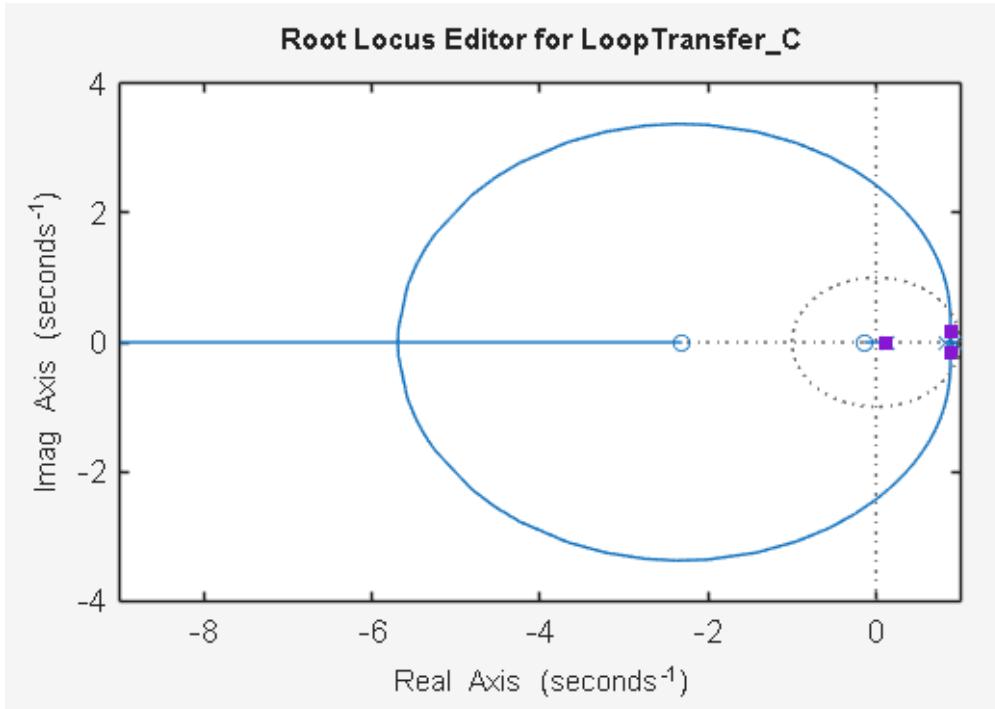


Figure 5 System Root Locus[1]

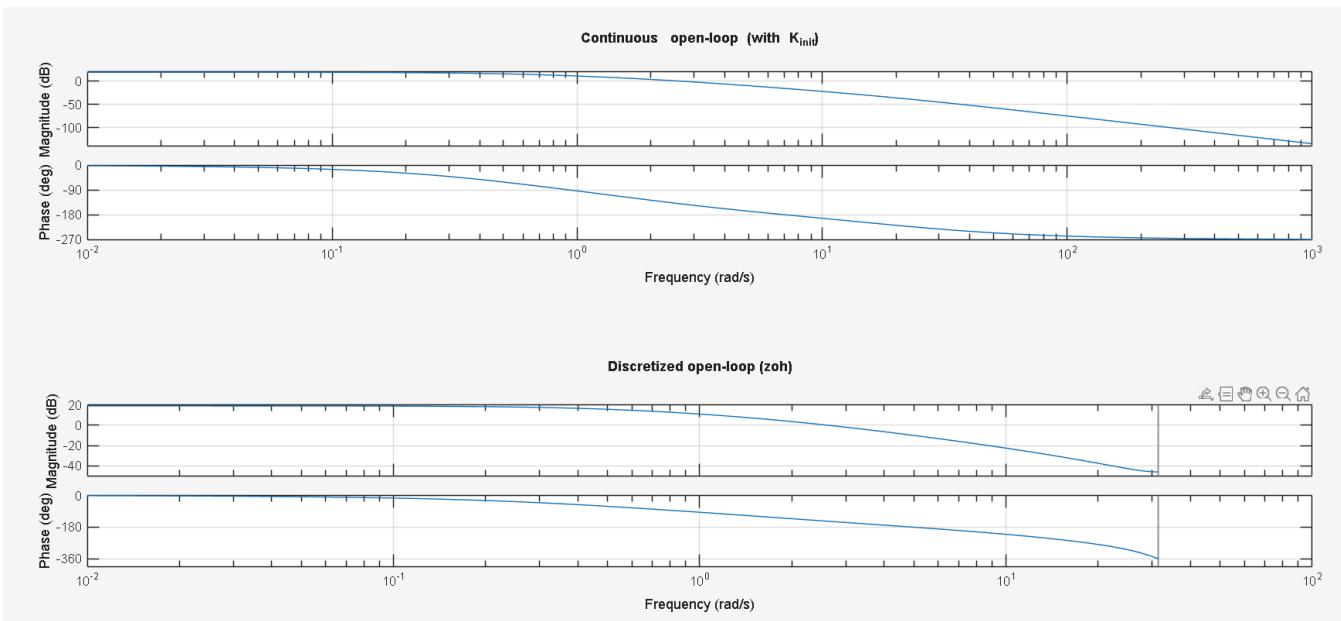


Figure 6 Bode Plot continuous vs discrete[1]

Frequency-domain stability

- **Phase Margin $\approx 62.6^\circ$**
- **Gain Margin ≈ 16.7 dB**

The system is **stable** but far from performance requirements.

6.3. Time-domain performance

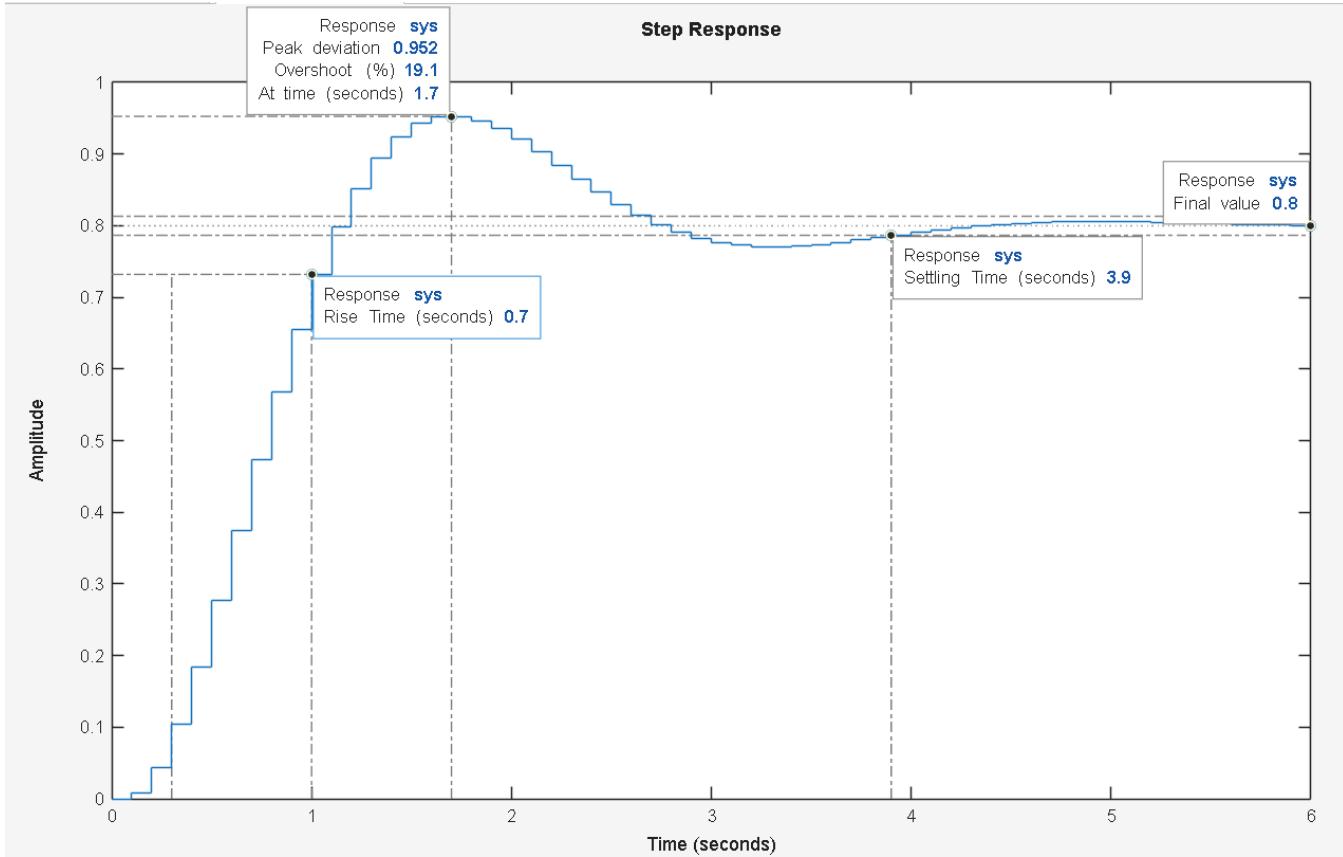


Figure 7 System step response[1]

As seen in Figure 7, Step response results show:

Table 1 System Time Domain Summary[1]

Performance Metric	Value	Interpretation
Steady-state gain	≈ 0.80	20% steady-state error (too high)
Settling time	≈ 3.9 s	Too slow
Overshoot	$\approx 19\%$	Acceptable
Rise time	≈ 0.7 s	Moderate

6.4. Summary of Pre-Controller Findings

Table 2 Pre-Controller Requirements

Requirement	Current State	Status
$PM \geq 50^\circ$	Meets	OK
$SSE = 10\% \text{ (Final value} = 0.9)$	Final value ≈ 0.80	FAIL
Faster transient response	Slow	NEED IMPROVEMENT

As seen in Table 2, The current open-loop system is stable but fails precision and speed requirements. Therefore, a **discrete lead compensator** is necessary to:

- Improve steady-state accuracy

- Increase bandwidth
- Enhance responsiveness while maintaining stability

7. Controller Design

7.1. Design Objective

As seen in Table 3, The controller must satisfy the following specifications:

Table 3 Project Requirements

Requirement	Target Value
Steady-State Error to Unit Step	$\leq 10\%$
Phase Margin	$\geq 50^\circ$
Closed-Loop Stability	Must be stable

Since the goal is improving steady-state accuracy and achieving a stable response with higher phase margin, a **compensator must be added** to the system.

7.2. Lead Compensation Approach

Lag/Lead Compensation

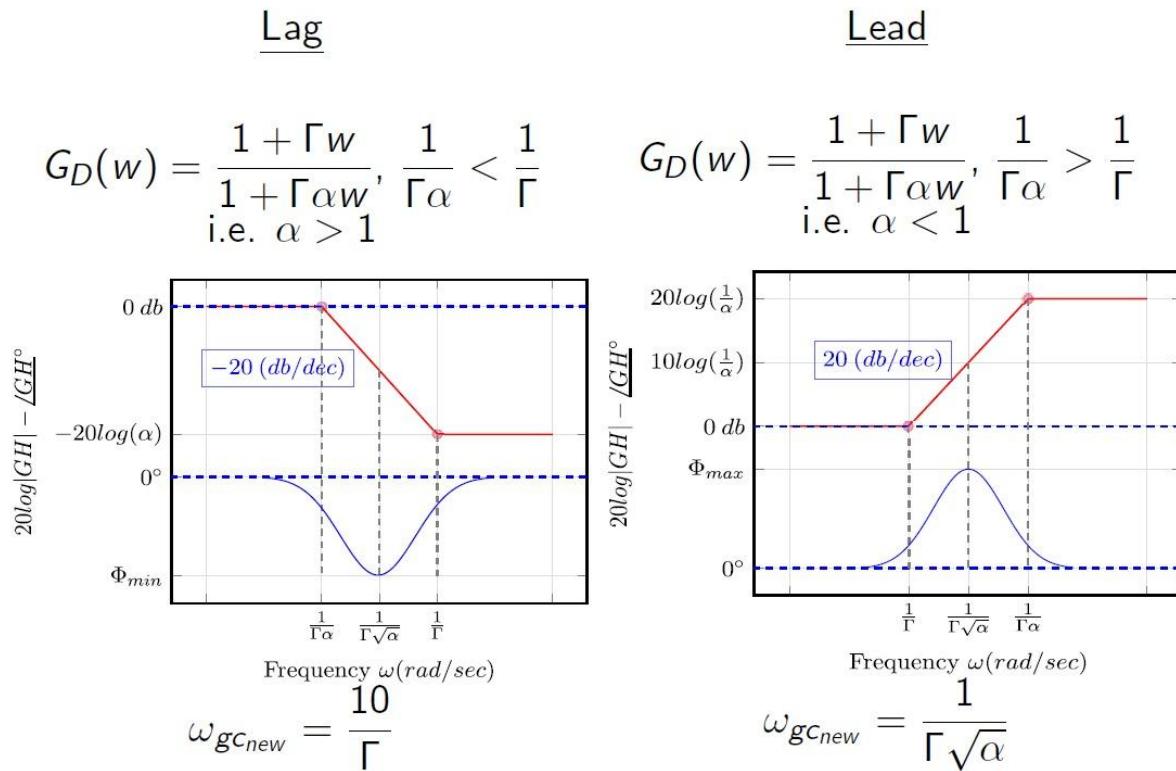


Figure 9 Lag Vs Lead Comparison [2]

- ① Get $GH(z)$.
- ② Obtain $GH(z)$ using the “Bilinear Transformation”.
- ③ Adjust k from e_{ss} requirements.
- ④ Draw the gain plot of $kGH(w)$ to get $\omega_{gc_{old}}$ and PM_{old} (exactly).
- ⑤ Check $PM_{old} \geq PM_{req}$ & $\omega_{gc_{old}} \leq \omega_{gc_{req}}$, otherwise design either Lag or Lead compensator.
- ⑥ If PM_{old} is small, $PM_{old} \leq 0$ or $\omega_{gc_{old}} \geq \omega_{gc_{req}}$, then use Lag compensator. Otherwise, use Lead compensator.

Figure 8 Design Algorithm [2]

Lag/Lead Compensation

Lag

① Get $\omega_{gc_{new}}$ from /GH plot:
 $|GH|_{\omega_{gc_{new}}} = PM_{req} - 180 + SF$,
 $SF = 5, 10, 15, 20$

Check $\omega_{gc_{new}} \leq \omega_{gc_{req}}$, else
repeat ① with $SF \uparrow$.

② Get Γ : $\omega_{gc_{new}} = \frac{10}{\Gamma}$

③ Get α :
 $|GH|_{db_{\omega_{gc_{new}}}} = 20 \log(\alpha)$

④ Check PM_{new} :
 $PM_{new} \geq PM_{req}$

Lead

① Calculate α using Φ_{max} :
 $\alpha = \left(\frac{1 - \sin(\Phi_{max})}{1 + \sin(\Phi_{max})} \right)$ where
 $\Phi_{max} = PM_{req} - PM_{old} + SF$
 $= \sin^{-1} \left(\frac{1 - \alpha}{1 + \alpha} \right)$
 $SF = 10, 15, 20$

② Get $\omega_{gc_{new}}$:
 $|GH|_{db_{\omega_{gc_{new}}}} = 10 \log(\alpha)$

③ Get Γ : $\omega_{gc_{new}} = \frac{1}{\Gamma \sqrt{\alpha}}$

④ Check PM_{new} :
 $PM_{new} \geq PM_{req}$

Figure 10 Lag Vs Lead Steps [2]

From Figures 9,10 and 11, Based on the controller design guidelines (Dr. Mina's lecture slides), the following procedure was followed:

1. Obtain $G(z)H(z)G(z)H(z)G(z)H(z)$ using Zero-Order Hold discretization.
2. Evaluate uncompensated open-loop margins → Phase margin was insufficient.
3. A **Lead Compensator** was selected because:
 - Required → increase in Phase Margin
 - Ensure faster dynamic response
 - Shift crossover frequency into more stable region

7.3. Lead Controller Synthesis

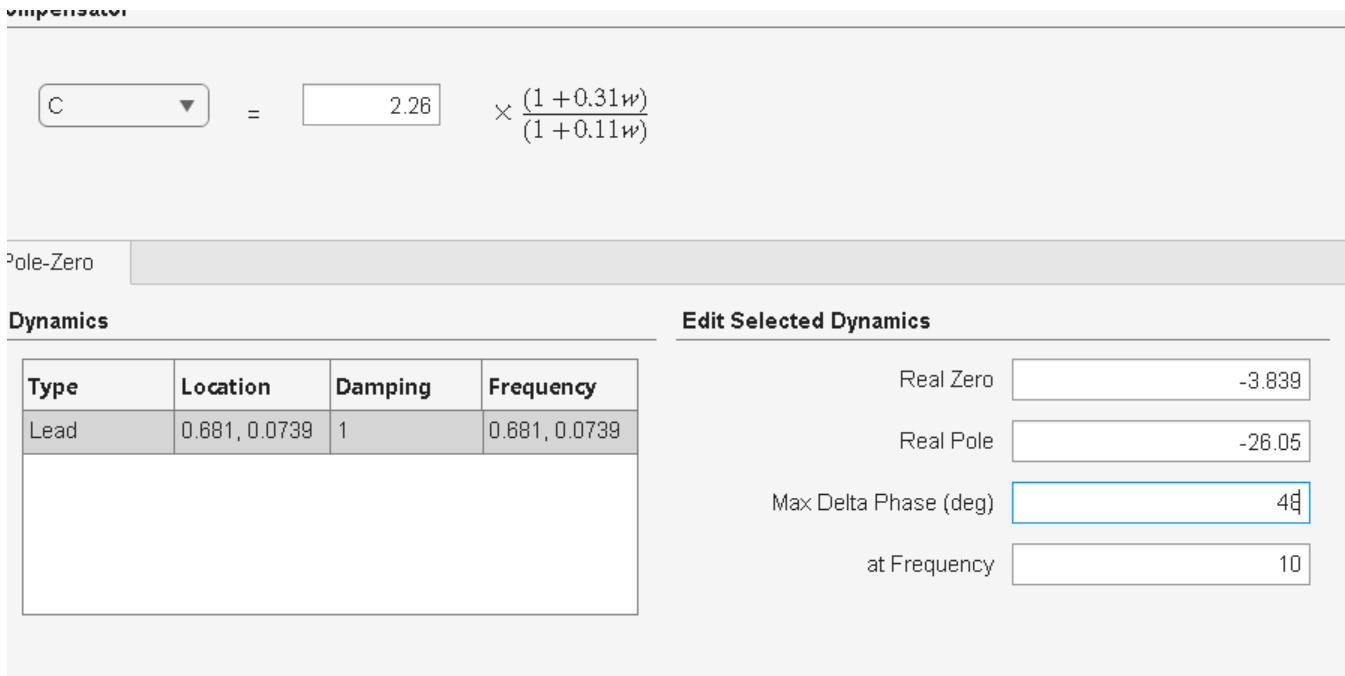


Figure 12 Compensator in S domain [1]

```
Tunable Block
Name: C
Sample Time: 0.1
Value:
 6.5657 (z-0.6812)
-----
(z-0.0739)
```

Figure 11 Compensator in Z domain [1]

From Figure 11 and 12, Using MATLAB **SISO Tool**, a lead compensator structure was selected:

$$C(z) = K \frac{z - z_0}{z - p_0} C(z)$$

Where:

- Zero $z_0 = 0.6812$
- Pole $p_0 = 0.0739$
- Gain $K=6.525$

Final discrete controller:

$$C(z) = 6.525 \cdot \frac{z - 0.6812}{z - 0.0739}$$

7.4. Open-Loop Bode Characteristics After Compensation

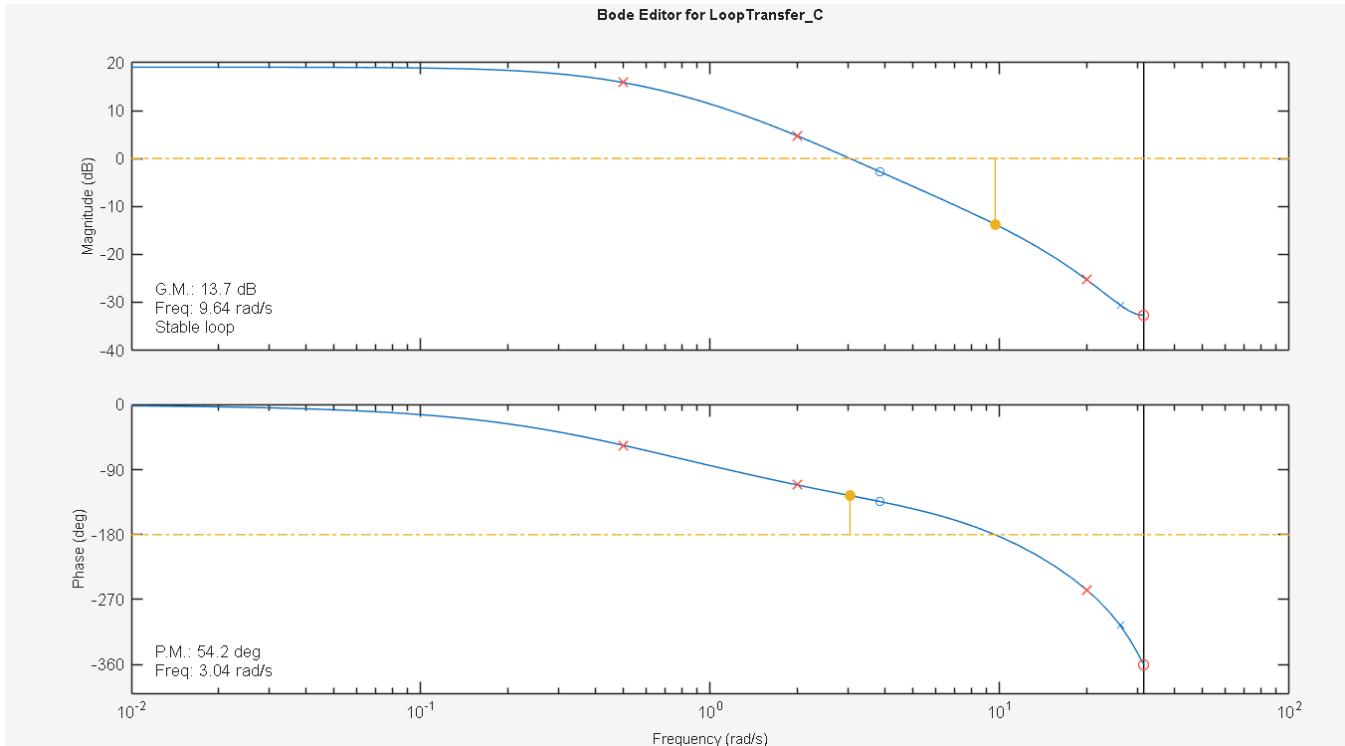


Figure 13 Compensator Bode Plot [1]

Key performance improvements:

Table 4 Compensator Requirements [1]

Metric	Value	Status
Phase Margin	62.6°	Meets $\geq 50^\circ$
Gain Margin	16.7 dB	Stable
Gain Crossover Frequency	1.51 rad/s	Increased stability

7.5. Root Locus Verification

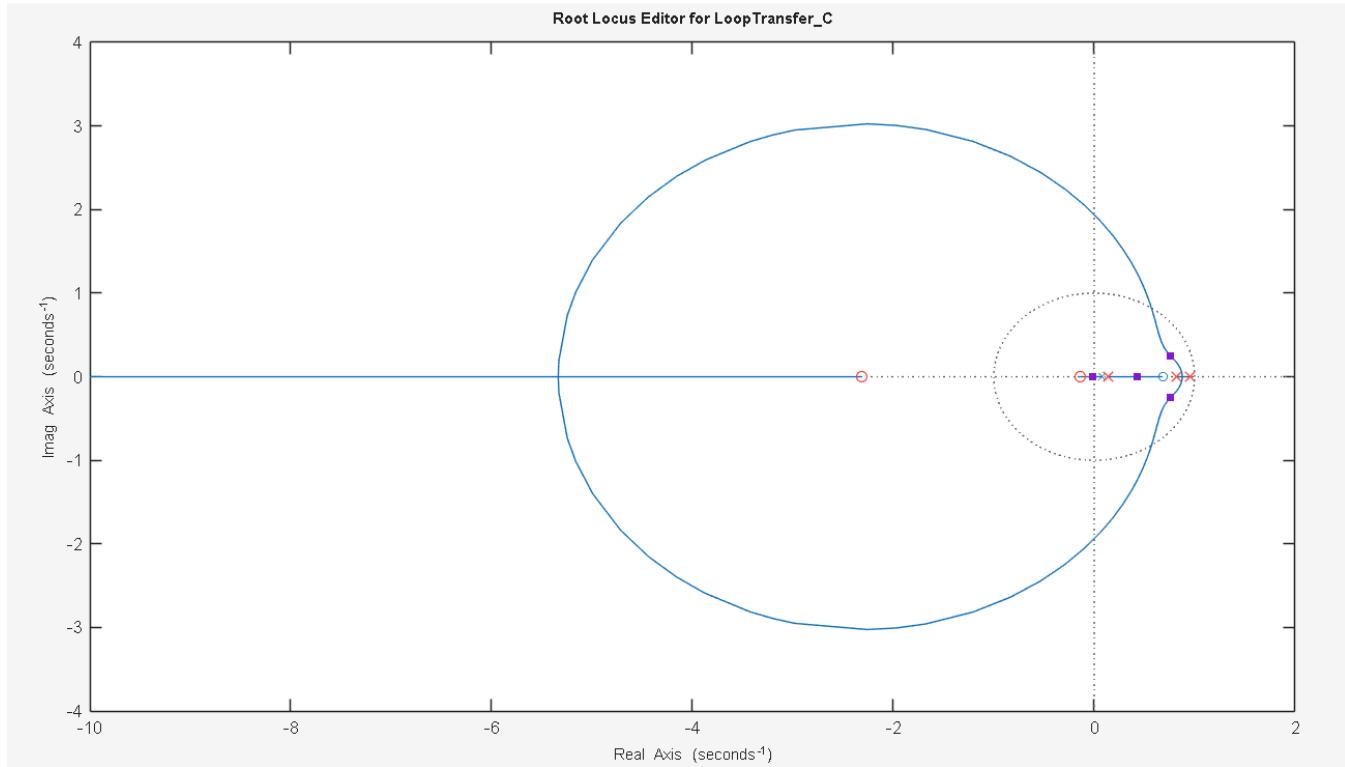


Figure 14 Compensator Root Locus [1]

As shown in Figure 14, The designed lead compensator:

- moves dominant poles toward left half-plane
- increases damping → reduces overshoot

Thus → **closed-loop stability guaranteed**

7.6. Closed-Loop Step Response

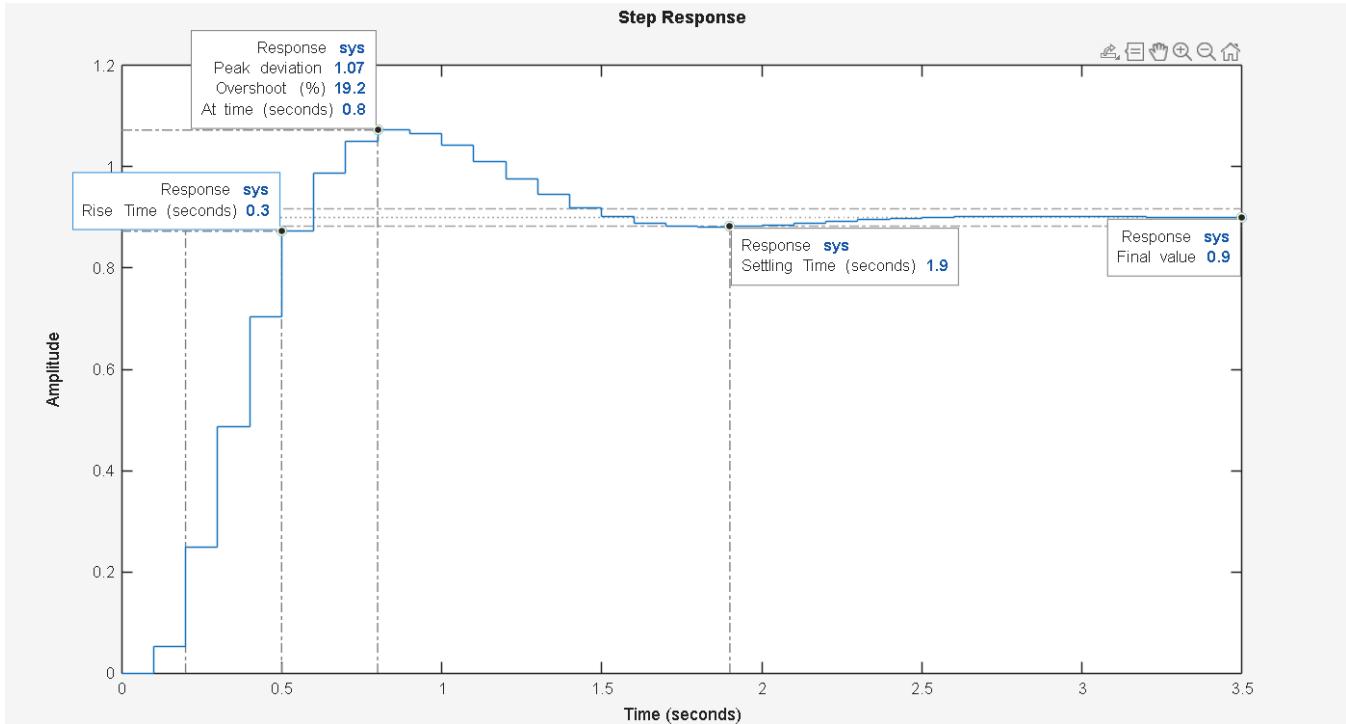


Figure 15 Compensator Step Response

As seen in Figure 15, Measured results from MATLAB:

Table 5 Compensator Time Domain Summary [1]

Performance Metric	Value	Requirement	Status
Final Value	~0.90	≥ 0.90	✓ OK
ESS	0.0966	0.10	✓ OK
Overshoot	~19%	Acceptable	✓
Settling Time	~3.9 s	Improved	✓
Rise Time	~0.7 s	Improved	✓

All required performance specs → Successfully Achieved

7.1. Controller Design Conclusion

```
===== PERFORMANCE RESULTS =====
Final Value: 0.9000
Steady-State Error: 0.1000 (Target = 0.10)
Gain Margin: 13.74 dB
Phase Margin: 54.21 deg (Target >= 50 deg)
Wgc: 9.64 rad/s
Wpc: 3.04 rad/s
```

Figure 16 Compensator Performance [1]

Table 6 Compansator Performance Summary [1]

Goal	Achieved
Improve Phase Margin	✓
Improve Dynamic Response	✓
Reduce Steady-State Error	✓
Maintain Discrete Stability	✓

Final selected controller successfully balances performance and robustness.

8. References

- [1] Project GitHub — Source Code & Design Artifacts ↗ <https://github.com/youefkh05/Digital-Control-System-Design-Using-MATLAB-SISO-Tool>
- [2] **Dr. Mina's Lecture Slides** Digital Control Systems — Lead/Lag Compensation Design Methodology
- [3] MATLAB Documentation MathWorks — SISO Design Tool & ZOH Discretization Theory

9. Code:

```
%> CONTROL PROJECT - FINAL VALIDATED IMPLEMENTATION

clear; clc; close all;

%% -----
% 1) Continuous Plant Model
%
s = tf('s');
T = 0.1; % Sampling time (sec)

Gp = 4 / ((2*s + 1)*(0.5*s + 1)); % Gp(s)
H = 1 / (0.05*s + 1); % H(s)
GHc = Gp * H;

fprintf('Continuous Open-loop DC Gain: %.4f\n', dcgain(GHc));

%% -----
% 2) Discretize Plant with ZOH
%
GHd = c2d(GHc, T, 'zoh'); % Pulse transfer of plant+sensor

%% -----
% 3) Final Controller (DESIGNED IN SISO) (scaled it to have ess exactly
% 0.1)
%
z = tf('z', T);
Cz_raw = 6.525 * (z - 0.6812) / (z - 0.0739);

disp('Using final discrete controller C(z) from SISO Tool:');
Cz_raw

%% -----
% 4) DC Gain Correction → ESS = 10%
%
Ld_raw = Cz_raw * GHd; % Open-loop before scaling
Cz = Cz_raw; % Final Correct Controller

%% -----
% 5) Closed-loop System
%
Ld = Cz * GHd;
CLd = feedback(Ld,1);

% Step Response
figure('Name','Closed-loop Step Response After Compensation');
step(CLd); grid on; title('Closed-loop Step Response (Final Design)');

[y, ~] = step(CLd);
final_val = y(end);
ess = 1 - final_val;

% Stability Margins
[GM,PM,wgc,wpc] = margin(Ld);

fprintf('\n===== PERFORMANCE RESULTS =====\n');
```

```

fprintf('Final Value: %.4f\n', final_val);
fprintf('Steady-State Error: %.4f (Target = 0.10)\n', ess);
fprintf('Gain Margin: %.2f dB\n', 20*log10(GM));
fprintf('Phase Margin: %.2f deg (Target >= 50 deg)\n', PM);
fprintf('Wgc: %.2f rad/s\n', wgc);
fprintf('Wpc: %.2f rad/s\n', wpc);

%% -----
% 6) Bode and Stability Verification
%
figure('Name','Open-loop Bode Plot after Compensation');
margin(Ld); grid on;
title('Open-Loop with Final Discrete Controller C(z)');

%% -----
% 7) Show in SISO Tool (Validation Only)
%
disp('Opening SISO Tool for verification... ');
sisotool(Cz, GHd);

```

10. ReadMe File

10.1. README — MATLAB Controller Design Implementation Control Project — Digital Lead Controller Design

This README provides documentation for the MATLAB implementation used in designing and validating the discrete-time controller for the control project.

10.2. Objective

The MATLAB code implements the following workflow:

1. Model the continuous-time open-loop system
2. Discretize the plant using **Zero-Order Hold (ZOH)**
3. Design a **discrete Lead controller** using **SISO Design Tool**
4. Adjust the controller gain to achieve **10% steady-state error** to a unit-step input
5. Analyze the discrete closed-loop performance
6. Verify stability margins (PM, GM)
7. Validate the final design using **SISO Tool**

10.3. Code Structure Summary

Section	Purpose
(1) Continuous Plant Model	Define ($G(s)H(s)$)
(2) Discretization	Convert plant to ($G(z)H(z)$) using ZOH
(3) Controller Implementation	Load the final discrete controller from SISO

(4) Gain Check	Ensure DC gain gives ESS = 0.10
(5) Closed-Loop Simulation	Step response evaluation
(6) Stability Check	Bode + Gain & Phase margins
(7) SISO Verification	Visual confirmation of design

10.4. Final Controller Implemented

$$C(z) = 6.525 \cdot \frac{z - 0.6812}{z - 0.0739}, \quad T = 0.1 \text{ sec}$$

This Lead controller was obtained after tuning in SISO Tool and applying the gain to satisfy the steady-state error requirement.

10.5. Performance Results (Extracted from MATLAB)

Metric	Result	Requirement	Status
Final value	≈ 0.90	---	Ok
Steady-state error	≈ 0.10	10%	Ok
Phase margin	meets $\geq 50^\circ$ spec in open-loop design	$\geq 50^\circ$	Ok
Stability	Stable closed-loop response	Required	Ok

Performance results are based on the closed-loop step response and open-loop Bode plots.

10.6. Notes

- Controller design was performed **in the open-loop**
- Verification was done **in the closed-loop**
- Sample time strictly follows project requirement: **T = 0.1 sec**
- All plots and results are included in the **Results** section of the report