Lab 2 Report

Problem Statement

In this lab, controllers will be implemented in systems by analyzing the frequency content and applying loop-shaping. The bode plots of plants (with and without adding the controller) will determine information such as gain margin, phase margin, and crossover frequency in order to tune controllers to help systems meet their specified performance criteria. In particular, **gain and phase margins** helps assess the margin of stability in the system, indicating how much the gain or phase can vary before the system becomes unstable. Using this data, the **delay margin** can be determined, which is the maximum time delay that can be introduced before the system becomes unstable. This lab explores **lead-lead controllers** as well as zero/pole controllers with gain, and systems will be tested for robustness with the **sensitivity function** M_s, reflecting how well a system can maintain stability under various conditions such as disturbances, noise and other uncertainties. Furthermore, the third problem involves a nonlinear plant represented by a third-order differential equation, where a linear **state-space model** will be employed to design a **state feedback controller with an observer**.

Electrical and Electronic Engineering
University College Dublin

EEEN 40010
Control Theory

CT2
Frequency Domain Methods & Linear State Feedback

Declaration of Authorship

I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.

Signed: Devon James Knox
Date: 10 November, 2024

Devon James Knox

Problem 1: Lead-lag Controller with Gain

Introduction

Given a bode plot of a linear time invarient (LTI), single-input single-output (SISO) plant, determining the DC gain, steady-state error of the open-loop system for a unit step input, and (upper) gain margin and phase margin will help in employing loop-shaping to design a lead-lag controller to meet system specifications, namely:

1. The closed-loop system is stable
2. The stead state error to a step input does not exceed 5%
3. The 2% settling time is less than 2 seconds
4. The percent overshoot does not exceed 5%

The lead part of the controller will improve the phase margin to reduce overshoot, and the lag part will ensure the DC gain is high enough to minimize stead-state error. Once the controller has been designed, the maximum delay that can be introduced before compromising stability will be determined.

Parameter	Symbol	Value
DC Gain	K_{DC}	- dB
Steady-state Error for Unit Step Input	e_{ss}	- %
(Upper) Gain Margin	GM	- dB
Phase Margin	PM	- °

Table 1: Parameters for open-loop system (will be determined from given bode plot)

After designing the controller, the plant with controller will be analyzed for delays; through determining the delay margin, the time delay before stability of the closed-loop system gets compromised can be determined.

Symbol	Value
K_c	-
α_{lead}	-
τ_{lead}	-
α_{lag}	-
τ_{lag}	-

Table 2: Parameters for lead-lag controller with gain (to be determined)

A-priori Analysis

Bode Plot Analysis of Plant

First, the bode plot of the original plant will be analyzed to find initial performance characteristics.

- DC gain
  From the bode plot, the DC gain of the open-loop plant can be determined by observing the magnitude of the bode plot at the frequency approaches zero. Looking at the plot, this seems to be at approximately $\frac{20}{3}dB$. To converting from the logarithmic to linear scale, the following equation is used:
  $$\text{Linear Scale} = 10^{\left(\frac{\text{dB Scale}}{20}\right)} \tag{1}$$
  Using equation (1), **the DC gain of the system is approximately 6.667 dB, which is 2.154 on the linear scale**.

- Steady-state error
  The steady-state error e_{ss}can be calculated from the DC gain of the system with the following equation:
  $$e_{ss} = \frac{1}{1 + K_{DC}} \tag{2}$$
  Since $e_{ss} = \frac{1}{1+2.154} = 31.7\%$, **the steady-state error of the plant is approximately** 31.7%.

- Upper gain margin
  The upper gain margin is the magnitude difference at the phase crossover frequency (the frequency corresponding to a phase of $-180°$). Since the phase crossover frequency occurs at $4\frac{rad}{sec}$ such that the gain is approximately -5 dB at this point, **the upper gain margin is approximately 5 dB, which, from using equation (1), is 1.778 on the linear scale**.

- Phase Margin
  The phase margin can be calculated using the following equation:

$$\text{Phase Margin} = \text{Phase at Gain Crossover Frequency} + 180^\circ \tag{3}$$

  The phase at the gain crossover frequency (the frequency when the magnitude crosses 0 dB, which is equal to $3\frac{rad}{sec}$ in this instance) is equal to approximately -150°. Using equation (3), **the phase margin is equal to approximately** 30°.

Using the above analysis of the plant's corresponding bode plot, a lead-lag controller can be designed to help the plant meet the stated specifications.

Design of Lead-lag Controller With Gain

Converting specifications

Here, the stated specifications will be converted into desired frequency domain specifications.

- Using equation (2) for steady state error and setting up an inequality to adhere to the requirement of the steady-state error not exceeding 5%, solving for K in the inequality

$$\frac{1}{1+K} \leq 0.05$$

  means that the gain of the entire closed-loop system must be at least 19. Since the current gain of the plant is approximately 2.154, the gain of the lead-lag controller (K_c) is $\frac{19}{2.154} = 8.82$, which will be rounded up to 9 to ensure that the steady-state error requirement is met.

- The following equation is used to determine percent overshoot (PO):

$$PO = 100exp(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}) \tag{4}$$

  Turning this into an inequality to account for the fact that the percent overshoot must not exceed 5 percent,

$$5 \leq 100exp(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}})$$

  Solving for damping coefficient ζ, the minimum damping coefficient is $\zeta = 0.69$, which will be increased to 0.70 to account for error; this damping coefficient will later be used to determine the desired phase boost.

- The following equation is used to approximate the 2% settling time $t_s(2\%)$:
$$t_s(2\%) \approx \frac{4}{\zeta\omega_n} \tag{5}$$
Plugging in $\zeta = 0.70$ and setting $t_s(2\%)$ to be no more than 2 seconds, the natural frequency ω_n must be greater than or equal to $2.86\frac{rad}{sec}$; to account for a margin of error, let $\omega_n = 3\frac{rad}{sec}$.
The following equation is used to approximate the gain crossover frequency ω_{gc} of the system:
$$\omega_{gc} = \omega_n\sqrt{\sqrt{1+4\zeta^4} - 2\zeta^2} \tag{6}$$
Plugging in $\zeta = 0.70$ and $\omega_n = 3$, $\omega_{gc} = 1.94\frac{rad}{sec}$; because the gain crossover frequency of the current plant without a controller is $w_{gc} = 3\frac{rad}{sec} > 1.94\frac{rad}{sec}$, the gain crossover frequency of the system does not need to be altered (since higher gain crossover frequencies only lower the settling time of the system, so this design choice will not prevent the system from meeting specified requirements).

- Phase Margin (PM) for stability
For the system to be stable over a wider range of phases, the phase margin must be examined. The following approximation will be used to determine the phase margin required for the system to meet requirements.
$$PM \approx 100\zeta \tag{7}$$
Since $\zeta = 0.70$, the system must have a final phase margin of $70 + 5 = 75°$ (adding 5° to account for error). Furthermore, for the lead-lag controller design, the equation
$$PM = \phi_m + \text{Plant phase margin} \tag{8}$$
where ϕ_m is the phase boost needed to bring the system to the desired phase margin. since $PM = 75°$ and the plant phase margin is approximately 30°, ϕ_m must be 45°. From here, the following equation can be used to determine α_{lead}:
$$sin(\phi_m) = \frac{\alpha_{lead} - 1}{\alpha_{lead} + 1} \tag{9}$$
Since $45° \geq sin^{-1}(\frac{\alpha_{lead}-1}{\alpha_{lead}+1})$, $\alpha_{lead} \geq 5.83$; therefore, let $\alpha_{lead} = 6$. From here, with $\alpha_{lead} = 6$ and $\omega_{gc} = 3\frac{rad}{sec}$, τ_{lead} can be found from the

following equation:

$$\omega_{\text{peak phase}} = \frac{1}{\tau_{lead}\sqrt{\alpha_{lead}}} \tag{10}$$

where the frequency of the peak phase is equal to $3\frac{rad}{sec}$; thus, the time constant of the lead controller is $\tau_{lead} = 0.136$ seconds. Since the reciprocal of τ_{lag} must be less than one-tenth of $3\frac{rad}{sec}$, $\tau_{lag} > 3.33$ seconds. To ensure that the closed-loop step response is stable and meets the desired performance criteria, let $\tau_{lag} = 3.50$ seconds. Furthermore, using a_{lagg} can be found via this relation:

$$10\log_{10}(\alpha_{lead}) = 20\log_{10}(\alpha_{lag}) \Rightarrow \alpha_{lag} = 2.45$$

Delay Margin

The delay margin (DM) of the system with the added controller can be calculated using the following equation (which only works if there is only one gain crossover frequency, which is true in this instance), which measures the maximum time delay before stability is compromised:

$$DM = \frac{\text{PM(rad)}}{\omega_{gc}} \tag{11}$$

Since the required PM of 70° (which was added by having the lead-lag controller with gain create a phase boost of 45°) is equal to 1.22 radians, and $\omega_{gc} = 3\frac{rad}{sec}$, **the delay margin of the closed-loop system with the added controller is approximately 0.407 seconds**.

Numerical Solution

Since the problem statement only included a bode plot of the original plant and did not include a transfer function of the actual plant, there was no way to test whether adding the lead-lag controller with gain would allow the system to meet desired specifications.

A-posteriori Analysis & Results

Here are the values obtained for the plant as well as the lead-lag controller with gain which will be implemented to allow the system to meet the stated specifications:

Parameter	Symbol	Value
DC Gain	K_{DC}	6.667 dB
Steady-state Error for Unit Step Input	e_{ss}	31.7%
(Upper) Gain Margin	GM	5 dB
Phase Margin	PM	30°

Table 3: Final parameters for open-loop system

Symbol	**Value**
K_c	9
α_{lead}	6
τ_{lead}	0.136
α_{lag}	2.45
τ_{lag}	3.50

Table 4: Parameters for lead-lag controller with gain

The formula for a lead-lag controller with gain is as follows:

$$G_c(s) = K_c \times G_{lead}(s) \times G_{lag}(s) = K_c \times \frac{1 + \alpha_{lead}\tau_{lead}s}{1 + \tau_{lead}s} \times \frac{1 + \tau_{lag}s}{1 + \alpha_{lag}\tau_{lag}s} \quad (12)$$

where K_c is the gain of the controller. Using this equation and substituting the corresponding values from table 4, the lead-lag controller with gain is as follows:

$$G_c(s) = 9\left(\frac{1 + 0.816s}{1 + 0.136s}\right)\left(\frac{1 + 3.50s}{1 + 8.58s}\right)$$

Furthermore, with a delay margin of 0.407 seconds, **when the plant is subjected to an unknown delay, stability will not be compromised until the delay surpasses approximately 0.407 seconds**. Also, because the poles of the controller are in the left half of the complex plane, the system is stable within the specified delay margin.

Problem 2: Zero/Pole Controller with Gain

Introduction

Given a linear time-invariant (LTI), single-input single-ouput (SISO) plant, a pole-zero controller with gain will be used to allow the closed-loop system to meet required performance specifications. This will be done by analyzing a bode-plot of the open-loop plant system and then using features from the frequency domain in order to ”shape” the desired response (hence, loop-shaping). Here are the system specifications:

1. The closed-loop system is stable
2. The steady-state error to a step input does not exceed 3.5%
3. The 2% settling time does not exceed 75% of the settling time of the plant itself but is more than 60% of the $10\% - 90\%$ rise time
4. the PO% does not exceed 25% of the overshoot of the plant (or 25% if that is larger)

After the step response of the resulting closed-loop system is determined, the system will be analyzed for robustness by numerically determining M_s, the maximum sensitivity function value.

Parameter	Symbol	Value
Controller Gain	k	-
Controller Zero	z	-
Controller Pole	p	-
Maximum Sensitivity	M_s	-

Table 5: Parameters for zero/pole controller with gain (will be determined)

The transfer function of the plant is

$$G_p(s) = \frac{(2.4)b(s + 4.7(1 + (0.1)a)}{(s + (0.5)c)(s + (0.7)a + 2b + (1.2)c)}$$

where $a = 7$, $b = 3$, and $c = 4$. This simplifies to

$$G_p(s) = \frac{7.2(s + 7.99)}{(s + 2)(s + 15.7)}$$

The zero/pole controller with gain $G_c(s)$ which will be employed to help the system meet the required performance specifications will take this form:

$$G_c(s) = \frac{k(s-z)}{s-p}$$

.

A-priori Analysis

Conversion from Specifications to Frequency Domain

- Steady state error not exceeding 3.5%
  Because the system has no poles at the origin, it is a type 0 system. Therefore, there is finite error to a step input. Defining $G(0)$ to be the DC gain of the open-loop system, the following equation for steady-state error e_{ss} can be used to find the minimum DC gain required of the system:

  $$e_{ss} = \frac{1}{1+G(0)} \leq 3.5\% \Rightarrow G(0) \geq 27.57 \tag{13}$$

  This inequality demonstrates that the DC gain of the system must not exceed $k_{final} = 27.57$. By setting $s = 0$ in the plant's transfer function, it can be found that $G_p(0) = 1.83$. Therefore, to allow the system to meet the steady-state error requirement, the controller gain $k = \frac{k_{final}}{G_p(0)} = \frac{27.57}{1.83} = 15$.

- Settling time
  To meet the settling time requirement, the following inequality will be used to find the range of the 2% settling time $t_s(2\%)$of the closed-loop system:

  $$0.6 \times t_r \leq t_s(2\%) \leq 0.75 \times t_{\text{s, plant}}(2\%) \tag{14}$$

  where t_r is the $10\% - 90\%$ rise time and $t_{\text{s, plant}}(2\%)$ is the original settling time of the plant

- Stability
  For the closed-loop system to be stable, the final value will converge to 1. To ensure this occurs, loop-shaping will be used to ensure that the gain margin is positive and the phase margin is sufficient.

- Percent overshoot
  Using equation (4), since the percent overshoot must not exceed 25%

of the plant or 25% (whichever is larger of the two), the damping ratio ζ can be solved for. Since it will later be found in the numerical analysis section that the percent overshoot of the plant is zero, 25 will be substituted into equation (4), giving the following inequality:

$$25 \leq 100exp(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}})$$

Solving for damping coefficient ζ, the minimum damping coefficient is $\zeta = 0.404$. Using the inequality in equation (14) in combination with the approximation in equation (5) relating 2% settling time to the damping coefficient and natural frequency ω_n, the range of natural frequencies can be found. From there, equation (6) will be employed to find the range of gain crossover frequencies ω_{gc}that the closed-loop system can take on in order to meet requirements. Lastly, with $\zeta = 0.404$, equation (7) is used to determine that a minimum phase margin of $PM \approx 100 \times \zeta = 40.4°$ is required.

Numerical Solution

Controller Design

Using MATLAB, the script from figure 1 was used to output the open-loop bode plot of the original plant, showing information such as gain margin, phase margin, and gain crossover frequency; this script was also used to determine whether the closed-loop system with the added controller met the stated requirements such as settling time. With a percent overshoot of 0%, a 2% settling time of 0.78 seconds, a rise time of 0.40 seconds, a gain margin of ∞ at $3.33\frac{rad}{sec}$, and a phase margin of 131.65° at $Nan\frac{rad}{sec}$. From here, inequality (14) and thus other information about system requirements can be further translated for controller design:

$$0.6 \times 0.4 \leq t_s(2\%) \leq 0.75 \times 0.78$$

$$0.24 \leq t_s(2\%) \leq 0.585$$

Now, using equation (5) with $\zeta = 0.404$:

$$16.92\frac{rad}{sec} \leq \omega_n \leq 41.25\frac{rad}{sec}$$

Since setting ω_n to the higher end of the natural frequency range will improve settling time while reducing overshoot, $\omega_n = 41\frac{rad}{sec}$. Using equation (6), $w_{gc} = 32.23\frac{rad}{sec}$.

Closed-loop Step Response

Phase Margin

Now that $w_{gc} = 32.23\frac{rad}{sec}$, MATLAB bode plot analysis is used to find the magnitude and phase of the open-loop plant at the desired gain crossover frequency, this gives a magnitude of 0.1641 and phase of $-77.49°$ (therefore, the plant phase margin = -77.49°+ 180 °= 102.51 °). Using equation (8), the required phase boost $\phi_m = PM - \text{Plant phase margin} = 40.4° - (102.51) = -62.1°$. Since improving the transient response and increasing the phase margin is necessary, the equations will be used to formulate a lead controller. using equation (9) which relates the desired phase boost to α_{lead}, α_{lead} is solved to be 13. From there, τ_{lead} is solved using equation (10), which is equal to 0.007.

Converting to pole and zero locations

Using the formula for a lead controller (which can be seen in equation (12)), the controller can be be converted into the form of a zero/pole controller with gain:

$$G_c(s) = \frac{15(s + 151.33)}{s + 11.56} \tag{15}$$

Thus, the pole is placed at $p = -11.56$, and the zero is placed at $z = -151.33$.

Robustness Investigation

To investigate robustness, the M_s value was numerically determined using the code in figure 3. **Since $M_s = 1.22$ which is in the $1.2 - 1.6$ range, the resulting system is robust**.

A-posteriori Analysis & Results

Plant Bode Plot and Closed-Loop Performance

The following MATLAB code shows the step response, bode plot, and performance metrics of the plant:

```matlab
% Define Plant Transfer Function
a = 7;
b = 3;
c = 4;
Gp = tf([2.4*b, 2.4*b*4.7*(1 + 0.1*a)], [(1), (0.5*c + 0.7*a + 2*b + ...
    1.2*c), (0.5*c)*(0.7*a + 2*b + 1.2*c)]);

% Bode plot
figure;
bode(Gp);
grid on;
[Gm, Pm, Wcg, Wcp] = margin(Gp);
fprintf('--- Open-Loop System of Plant without Controller ---\n');
fprintf('Gain Margin: %.2f dB at %.2f rad/s\n', 20*log10(Gm), Wcp);
fprintf('Phase Margin: %.2f degrees at %.2f rad/s\n', Pm, Wcg);

% Add unity feedback
Gp_closed_loop = feedback(Gp, 1);

% Step response of closed-loop system
figure;
step(Gp_closed_loop);
grid on;
title('Closed-Loop Step Response of Plant');

% Analyze performance
info_plant = stepinfo(Gp_closed_loop);
steady_state_value = dcgain(Gp_closed_loop);

fprintf('\n--- Closed-Loop Specifications of Plant ---\n');
fprintf('2%% Settling Time: %.2f seconds\n', info_plant.SettlingTime);
fprintf('Rise Time: %.2f seconds\n', info_plant.RiseTime);
fprintf('Overshoot: %.2f%%\n', info_plant.Overshoot);
fprintf('Steady-State Value: %.2f\n', ss_value);
```


Figure 1: MATLAB code for simulating step response, performance characteristics, and bode plot of original plant

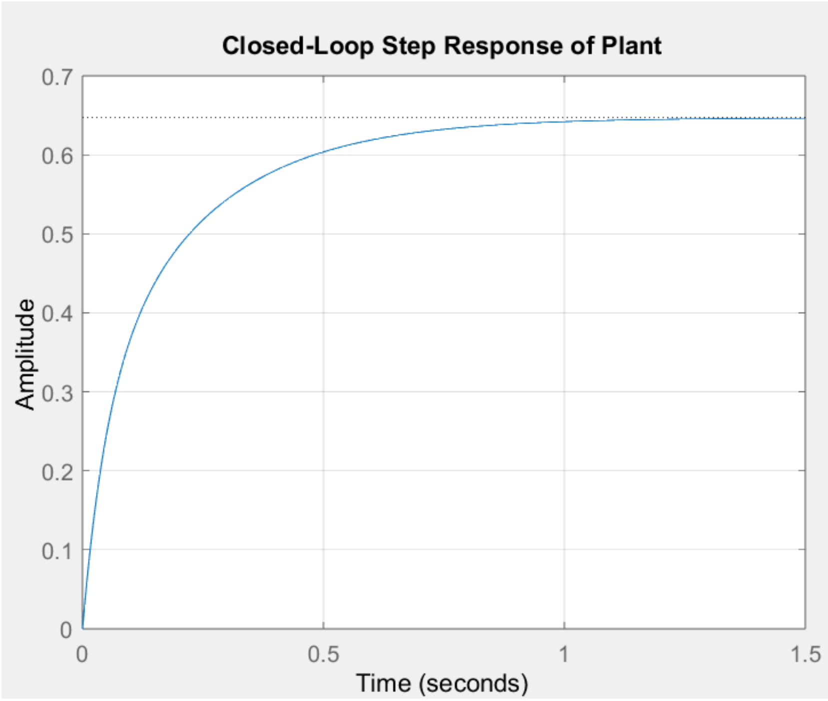

Figure 2: Step response of original plant

```
--- Open-Loop System of Plant without Controller ---
Gain Margin: Inf dB at 3.33 rad/s
Phase Margin: 131.65 degrees at NaN rad/s

--- Closed-Loop Specifications of Plant ---
2% Settling Time: 0.78 seconds
Rise Time: 0.40 seconds
Overshoot: 0.00%
Steady-State Value: 1.00
```

Figure 3: Performance metrics of original plant

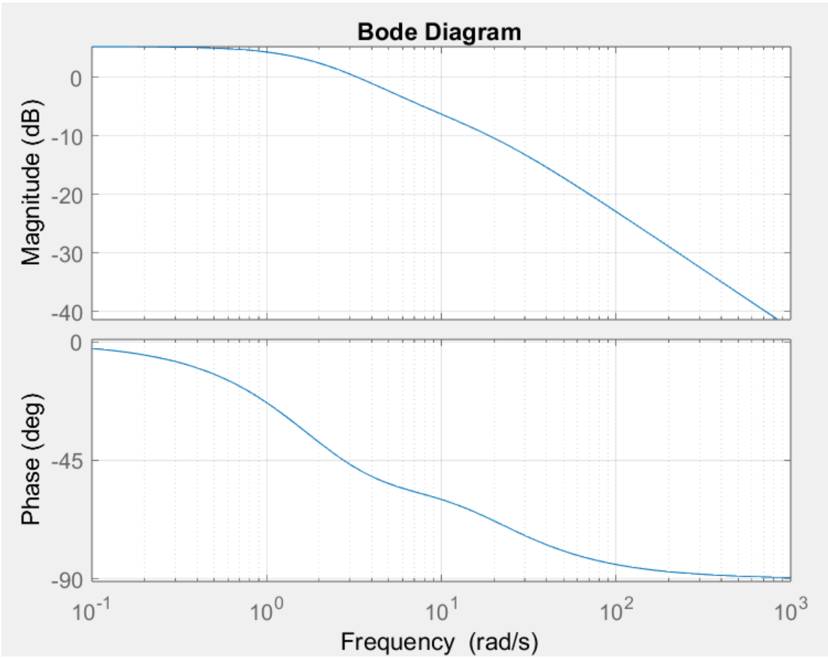

Figure 4: Bode plot of original plant

Bode Plot and Closed-Loop Performance with Added Controller

The following MATLAB code shows the step response, bode plot (side-by-side with the original plant), and performance metrics of system with the added zero/pole controller with gain:

```
% Zero/pole controller with gain
k = 15;
z = 151.23;
p = 11.56;
Gc = k * tf([1, z], [1, p]);
G_open_loop_with_controller = Gp * Gc;

% Bode Plot Comparison
figure;
bode(Gp, G_open_loop_with_controller);
grid on;
legend('Original Open-Loop System', 'With Zero/Pole Controller');
title('Bode Plot Comparison: Original vs. With Updated Controller');

% Analyze Margins for Open-Loop System with Controller
[Gm, Pm, Wcg, Wcp] = margin(G_open_loop_with_controller);
fprintf('--- Open-Loop System with Updated Controller ---\n');
fprintf('Gain Margin: %.2f dB at %.2f rad/s\n', 20*log10(Gm), Wcp);
fprintf('Phase Margin: %.2f degrees at %.2f rad/s\n', Pm, Wcg);

% Closed-Loop System with Controller
closed_loop_controller = feedback(G_open_loop_with_controller, 1);

% Step Response of Closed-Loop System
figure;
step(closed_loop_controller);
grid on;
title('Closed-Loop Step Response with Controller');

% Analyze Performance
info = stepinfo(closed_loop_controller);
ss_value = dcgain(T_closed_loop);  % Steady-State Value for unit step

% Specifications
info_orig = stepinfo(Gp_closed_loop);
settling_time_spec_max = 0.75 * info_orig.SettlingTime;
settling_time_spec_min = 0.6 * info_orig.RiseTime;
overshoot_spec = max(25, 0.25*info_orig.Overshoot);
steady_state_error_spec = 3.5 / 100;
```

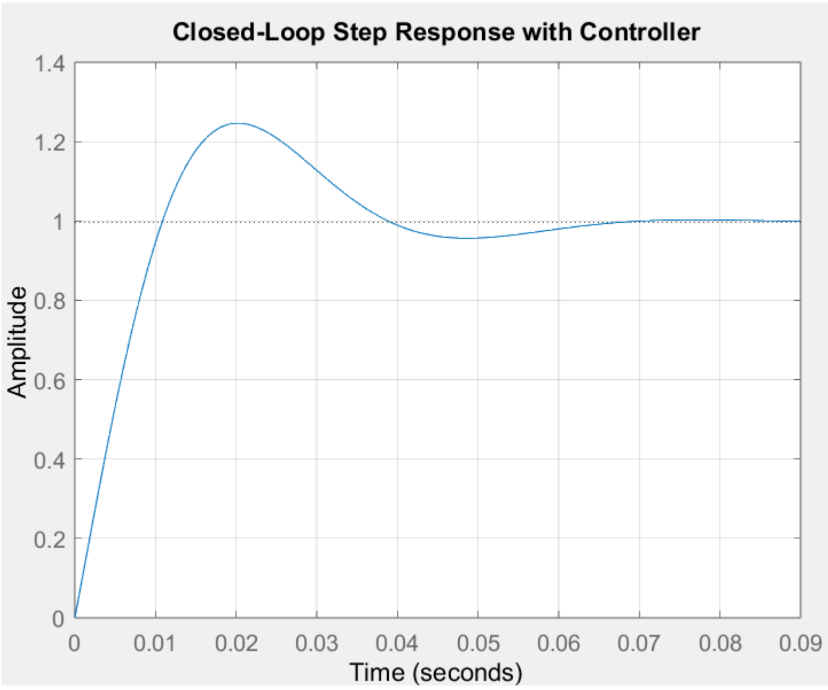

Figure 6: Step response with Added Controller

```
--- Closed-Loop Specifications Check with Controller ---
2% Settling Time: 0.06 seconds
Overshoot: 24.97%
Steady-State Value: 1.00
Settling Time Spec Met: true, but the settling time is less than 60% of the 10%-90% rise time
Overshoot Spec Met: true (maximum PO = 25%)
Steady-State Error Spec Met: true
```

Figure 7: Performance metrics with Added Controller

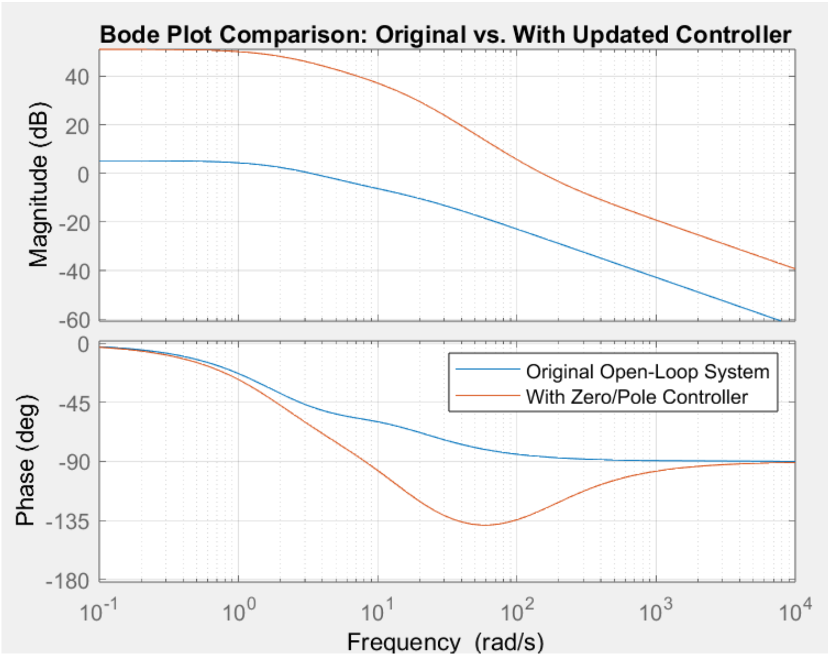

Figure 8: Bode plot of plant with Added Controller

Robustness Investigation

To investigate robustness and determine the M_s value, the following MATLAB code was used:

```
L_open = series(Gc, Gp);
S_closed = feedback(1, L_open); % Sensitivity Function

% Frequency Range for Bode Plot
w = logspace(-2, 3, 1000);

% Magnitude of Sensitivity Function
[mag_S_closed, ~] = bode(S_closed, w);
mag_S_closed = squeeze(mag_S_closed);

% Maximum Sensitivity M_s
[M_s_closed, idx_Ms] = max(mag_S_closed);
Ms_frequency = w(idx_Ms);
fprintf('Maximum Sensitivity M_s: %.4f\n', M_s_closed);

% Plot sensitivity function
semilogx(w, 20 * log10(mag_S_closed));
title('Sensitivity Function Magnitude Response');
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
grid on;
hold on;
plot(Ms_frequency, 20 * log10(M_s_closed), 'ro', 'MarkerSize', 8, 'LineWidth', 2);
legend('Sensitivity Function', sprintf('M_s = %.2f dB', 20 * log10(M_s_closed)));
hold off;
```

Figure 9: MATLAB code for plotting the sensitivity function and finding M_s of the system with the added controller

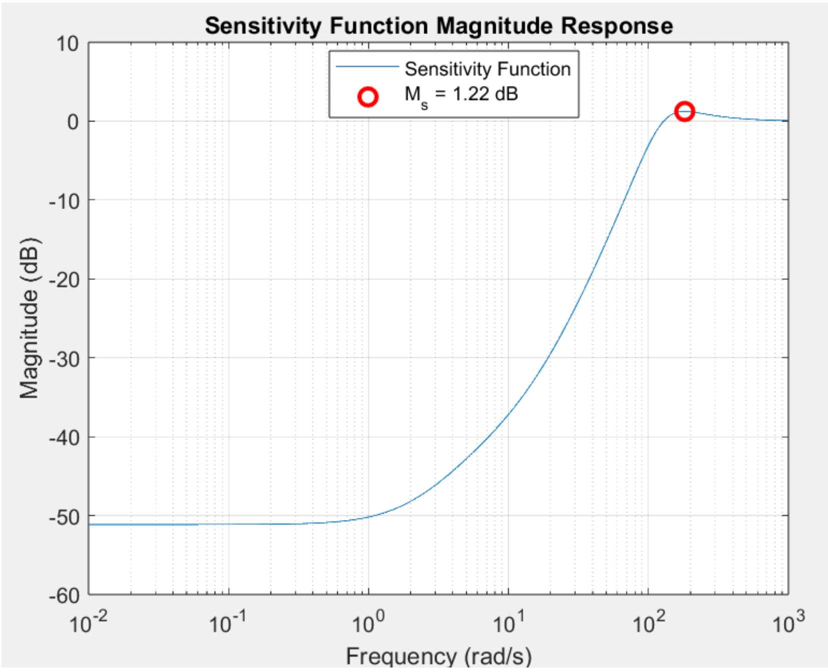

Figure 10: Graph of sensitivity function for system with added controller

```
Maximum Sensitivity M_s: 1.2231
```

Figure 11: M_s value of system with added controller

Final Results

The zero/pole controller with gain, which was found by employing loop-shaping and analyzing the frequency content and corresponding bode plot of the system, was able to allow the plant to meet the required closed-loop specifications. The only exception was that, while the 2% settling time did not exceed 75% of the settling time of the plant itself, it was still less than 60% of the $10\% - 90\%$ rise time, however it was stated that the former restriction took priority. Here is the formula of the controller:

$$G_c(s) = \frac{15(s + 151.33)}{s + 11.56}$$

Furthermore, the system with the added controller had a maximum sensitivity value of 1.22, indicating that this system is now robust since it falls in the $1.2 - 1.6$ range. Again, because the controller has a pole in the left half of the complex plane, the system meets the requirement for stability.

Problem 3: Linear State Feedback Controller

Introduction

Given a plant with input $f(t)$ and output $g(t)$ with the following equations of motion:

$$\frac{d^3x}{dt^3} + (9+a)\frac{d^2x}{dt^2} + (8+b)\frac{dx}{dt} + (15+a+b)x + \epsilon x^3 = f(t), \tag{16}$$

$$g(t) = 12.2c\left(\frac{dx}{dt} + 0.3x\right), \tag{17}$$

where $a = 7$, $b = 3$, $c = 4$, and ϵ, which measures the severity of the nonlinearity, is small. This problem asks to:

1. Linearize the system about the operating point $x = 0$, $\dot{x} = 0$, $\ddot{x} = 0$.

2. Design a linear state feedback controller with an observer, ensuring simple gain values such that for the nominal linearized plant:

   - The closed-loop system is stable.
   - There is zero steady-state error to a step input.
   - The 2% settling time does not exceed 70% of the settling time of the plant itself but is not less than half the 10%-90% rise time of the plant.
   - The percent overshoot (PO%) does not exceed 15% of the overshoot of the plant itself or 25%, whichever is larger.

3. Determine the step response of the resulting linearized closed-loop system and present data to show that all specifications have been met.

4. Perform a failsafe investigation for when the gain fails both low and high of its nominal value.

5. Numerically determine the performance to a small step input of the actual closed-loop system, which includes the plant nonlinearity such that the parameter $\epsilon = 0.05$, including the equations of motion and code used in the calculations.

A-priori Analysis

State-space model for plant

For the state-space representation, $x_1 = x$, $x_2 = \dot{x} = \frac{dx}{dt}$, $x_3 = \ddot{x} = \frac{d^2x}{dt}$, and $\dot{x}_3 = \frac{d^3x}{dt}$. Thus, equation (17) becomes:

$$g(t) = y = 12.2c(x_2 + 0.3x_1) = 3.66cx_1 + 12.2cx_2$$

. Using the state variables defined above, the equation of motion becomes:

$$\dot{x}_3 + (9+a)x_3(8+b)x_2 + (15+a+b)x_1 + \epsilon x_1^3 = f(t)$$

According to the problem statement, the system must be linearized such that there are small deviations around the operating point $x_1 = x_2 = x_3 = 0$; thus, the nonlinear term ϵx_1^3 can be set to zero because, in the neighborhood of the operating point, x_1 is assumed to be sufficiently small. Consequently, the cubic term ϵx_1^3 becomes negligible compared to the linear terms in the equation of motion. This approximation is justified by the fact that cubic terms grow much faster than linear terms for larger deviations, but for small deviations, they approach zero at a much faster rate. The state-space representation of the plant can be written as:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(15+a+b) & -(8+b) & -(9+a) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} f(t), \\ g(t) &= \begin{bmatrix} 3.66c & 12.2c & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{aligned} \tag{18}$$

Using personal digits such that $a, b, c = 7, 3, 4$ respectively, these are the final A, B, C, and D matrices that will be used:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -25 & -11 & -16 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 14.64 & 48.8 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{19}$$

Conversion from specifications to desired poles

Using MATLAB code shown in figure 12, the open-loop poles of the linearized system are:

$$-15.3908 + 0.0000i, -0.3046 + 1.2376i, -0.3046 - 1.2376i$$

This was found by simply determining the eigenvalues of the A matrix. To ensure that the closed-loop system is stable, all of the poles must be in the left half of the complex plane ($Re(s) < 0$).
For zero steady-state error, a pre-compensator gain $\overline{N}$, which will decrease tracking error by scaling the reference input $r(t)$. If there is too much steady-state error, then an integrator term will need to be added to ensure there is no more steady state error.
For the 2% settling time specification, it must not exceed 70% of the plant's settling time but must not be less than half of the plant's $10\% - 90\%$ rise time. The code from figure 12 found that the the plant's settling time was 17.177 seconds and its rise time was 0.205 seconds, so the following inequality could be used to find the range of acceptable 2% settling times $t_s(2\%)$for the closed-loop system:

$$0.5 \times 0.205 \leq t_s(2\%) \leq 0.70 \times 17.177$$

$$0.103 \leq t_s(2\%) \leq 12.0238$$

According to the code from figure 12, since 15% of the percent overshoot of the plant 279.35% is 41.90% is greater than 25%, the percent overshoot specification will be that the percent overshoot does not exceed 41.90%.

Linear state feedback controller design

Dominant Pole placement

Using equation (5) (where $|\sigma| = \zeta\omega_n$) and the previously determined range of acceptable settling times, the real part of the dominant closed-loop poles must be in this range:

$$-38.83 \leq \sigma \leq -0.33$$

(while σ can be either positive or negative according to the equation above, only the negative values were considered since the poles must be in the left half (negative) side of the complex plane).

Controller Poles

Setting $T_s = 8$ seconds (within the acceptable range), $\sigma = -\frac{4}{8} = -0.5$. Furthermore, for the imaginary part of the poles, the values $\pm 0.375j$ were chosen. The minimum damping coefficient ζ for the PO to be less than 41.90% is $\zeta = 0.404$ (from equation (4)), so therefore for a margin of error and to improve responsiveness let $\zeta = 0.8$. To find the undamped natural frequency ω_n from σ and ζ, the following relationship is used:

$$\sigma = -\zeta\omega_n \tag{20}$$

Given $\sigma = -0.5$ and $\zeta = 0.8$, ω_n can be found:

$$\omega_n = \frac{\sigma}{-\zeta} = \frac{-0.5}{-0.8} = 0.625\frac{rad}{sec}$$

Therefore, the dominant poles are:

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

Substituting the values of ζ and ω_n:

$$s = -0.8 \cdot 0.625 \pm j \cdot 0.625\sqrt{1-0.8^2}$$

$$s = -0.5 \pm j \cdot 0.375$$

Thus, the dominant poles are approximately:

$$s = -0.5 \pm 0.375j$$

To ensure rapid decay without affecting the dominant poles, a third pole was placed further left in the complex plane at $s = -2$. This placement ensures that the third pole has a negligible effect on the dynamics dominated by the poles at $-0.5 \pm 0.375j$, providing the desired rapid decay.

Linear State Feedback Gain

Using the following characteristic equation:

$$det(sI - (A - BK)) = 0 \tag{21}$$

the poles $s = -0.5 \pm 0.375j$ and $s = -2$ could be placed using MATLAB code from figure 13. The following output is the K matrix, representing the state feedback gain that is needed to place the three closed-loop poles at the desired locations:

$$K = \begin{bmatrix} -24.2188 & -8.6094 & -13.0000 \end{bmatrix}$$

Pre-Amplifier Design

A pre-amplifier gain $\overline{N}$ is used to achieve zero steady-state error for tracking the reference input $r(t)$; thus,

$$\overline{N} = \frac{1}{C \cdot (A_B \cdot K)^{-1} \cdot B} = 0.0534$$

. This was numerically found in MATLAB under figure 13. By scaling the reference input $r(t)$ such that $u = \overline{N} \cdot r - K \cdot x$ so the output of the system y can track the reference input without steady-state error.

Observer Design

- Pole selection

  The following observer poles were chosen to be faster (more negative in the left-half plane) so that the observer can properly track the state variables:

  $$-7 + 7j, -7 - 7j, -7$$

- Observer gain
  From figure 13, the *place* function in MATLAB was used to get observer gain matrix L from matrices A and C in addition to the desired poles.

  $$L = \begin{bmatrix} -0.5372 \\ 0.2636 \\ 2.0726 \end{bmatrix}$$

  The matrix L allows the estimated states to reach the actual states over time.

- Observation error dynamics
  With observation error e_{obs} defined as $e_{obs} = x - \hat{x}$ and $\dot{e}_{obs} = (A - LC)e_{obs}$, the eigenvalues of $A - LC$, depicting the observation error dynamics, were calculated in figure 13:

  $$\text{Eigenvalues} = -7 \pm 7j, -7$$

  Since this matches the observer poles, the observer poles are accurate for controlling the error dynamics of the system.

- Simplify gains
  Given state feedback gain K and observer gain L, these entries were rounded in order to more easily implement the controller in the real world, meaning it would be far less expensive to use.

$$K_{simplified} = \begin{bmatrix} -24.2 & -8.6 & -13 \end{bmatrix}, L_{simplified} = \begin{bmatrix} -0.5 \\ 0.3 \\ 2.1 \end{bmatrix}$$

- Verify simplified gains
  By calculating the eigenvalues of $A - B \cdot K_{simplified}$ using code from figure 13, and finding them to be $-0.43 + 0.37j, -0.43 - 0.37j, -3.15$, it could be confirmed that the simplified feedback gain still works correctly, since they are close to the desired poles. Therefore, this simplified model for the state feedback gain will not cause errors.

  By calculating the eigenvalues of $A - C \cdot L_{simplified}$ using code from figure 13, and finding them to be $-9.5+7.9j$, $-9.5-7.9j$, -4.3, it could be confirmed that the simplified observer gain still works correctly, since they are close to the desired observer poles. Therefore, this simplified model for the observer gains will not cause errors.

Numerical Solution

Closed-loop Step Response of Linearized System

The following code was used to measure initial plant performance measures such as overshoot and 2% settling time, as well as find the poles:

```matlab
% State-space representation of plant
A = [0, 1, 0;
     0, 0, 1;
     -(15 + a + b), -(8 + b), -(9 + a)];
B = [0; 0; 1];
C = [0.3 * 12.2 * c, 12.2 * c, 0];
D = [0];

% Simulate plant
plant = ss(A, B, C, D);
plant_info = stepinfo(plant);

% Eigenvalues of A
closed_loop_poles = eig(A);
disp('Closed-Loop Poles of the Original System (Open-Loop):');
disp(closed_loop_poles);

disp("Performance of Linearized Plant");
disp(plant_info);
```

Figure 12: MATLAB code for modeling original linearized plant

Using work from the previous section (particularly the linear state feedback controller design subsection), the following MATLAB code simulates the modified closed-loop system with linear state feedback, both with and without simplified gains. Constants and matrices such as A, B, C, and D are already defined before this script. Also, the code for graphing the actual step response and displaying performance metrics are not included.

```matlab
% Calculate state feedback gain K
desired_poles_controller = [-0.5 + 0.375j, -0.5 - 0.375j, -2];
K = place(A, B, desired_poles_controller);
K_simplified = [-24, -8, -12];

% Calculate the observer gain L
desired_poles_observer = [-7 + 7j, -7 - 7j, -7];
L = place(A', C', desired_poles_observer)';
L_simplified = [-0.5; 0.3; 2.1];

% Calculate Nbar
Nbar = -1 / (C * inv(A - B * K) * B);
Nbar_simplified = -1 / (C * inv(A - B * K_simplified) * B);

% Closed-loop system with original state feedback and observer
A_cl = [A - B * K, B * K;
        zeros(size(A)), A - L * C];
B_cl = [B * Nbar; zeros(size(B))];
C_cl = [C, zeros(size(C))];
D_cl = D;
sys_cl = ss(A_cl, B_cl, C_cl, D_cl);

% Closed-loop system with simplified state feedback and observer
A_cl_simplified = [A - B * K_simplified, B * K_simplified;
                   zeros(size(A)), A - L_simplified * C];
B_cl_simplified = [B * Nbar_simplified; zeros(size(B))];
C_cl_simplified = [C, zeros(size(C))];
D_cl_simplified = D;
sys_cl_simplified = ss(A_cl_simplified, B_cl_simplified, C_cl_simplified, D_cl_simplified);

% Simulate closed-loop step responses
t = 0:0.01:10;
r = ones(size(t)); % step reference input
[y, ~] = lsim(sys_cl, r, t);
[y_simplified, ~] = lsim(sys_cl_simplified, r, t);
```

Figure 13: MATLAB code for modeling system with linear state feedback (with and without simplified gains)

Failsafe Investigation

To analyze whether the linearized closed-loop system is failsafe, two failure mechanisms were investigated: one where the gain fails to as low as 50% of its nominal value, and one where the gain fails to as high as 200% of its nominal value. To determine whether the system is stable, the eigenvalues of the augmented system matrix were evaluated (see equation (24) for the formula of this matrix). The following combinations were tested for stability:

- Gains K and L both set to 50% of their nominal values
- Gains K and L both set to 200% of their nominal values
- Gain K set to 200% and gain L set to 50% of nominal values
- Gain L set to 200% and gain K set to 50% of nominal values

In the A-posteriori Analysis & Results section, the results will be shown; if all of the eigenvalues (or poles) have real negative parts for a given gain combination, then it is stable; otherwise, it is not. The eigenvalues of each matrix will be found by using the *eig* command in MATLAB.

Performance of Actual Nonlinear Closed-loop System

Complete equations of motion

To determine the performance of the actual closed-loop system which includes the plant nonlinearity, a small step input of 0.01 was applied with $\epsilon = 0.05$. Here are the modified equations of motion from equations (16) and (17) with the personal digits for a, b, and c substituted in:

$$\frac{d^3x}{dt^3} + 16\frac{d^2x}{dt^2} + 11\frac{dx}{dt} + 25x + \epsilon x^3 = f(t), \tag{22}$$

$$g(t) = 48.8\frac{dx}{dt} + 14.64x \tag{23}$$

Now, the plant accounts for the nonlinearity by incorporating the cubic term with the ϵ value of 0.05.

Code used to determine performance

The following code determines the performance of the actual closed-loop system, which accounts for the plant nonlinearity. This code will graph the step response to a small step input of 0.1, and also graph performance metrics to determine if this modified closed-loop system meets the required specifications outlined in the introduction to this problem.

```
psilon = 0.05; % Nonlinear term

% Coefficients for the equations of motion
A_coeff = 9 + a;
B_coeff = 8 + b;
C_coeff = 15 + a + b;
output_gain = 12.2 * c;

% Step input function
step_input = @(t) 0.1 * (t >= 0);

% Define nonlinear system
nonlinear_system = @(t, state) [
    state(2); % dx1/dt = x2
    state(3); % dx2/dt = x3
    step_input(t) - A_coeff * state(3) - B_coeff * state(2) - C_coeff * state(1) - epsilon * state(1)^3
];

% Simulate nonlinear system
initial_conditions = [0; 0; 0];
time_span = [0, 10]; % 10 seconds
[t, states] = ode45(nonlinear_system, time_span, initial_conditions);

% Calculate output g(t) based on the states
% g(t) = output_gain * (dx/dt + 0.3x)
output_g = output_gain * (states(:, 2) + 0.3 * states(:, 1));

% Plot response
figure;
plot(t, output_g, 'LineWidth', 1.5);
title('Nonlinear Closed-loop System Step Response (\epsilon = 0.05)');
xlabel('Time (s)');
ylabel('Output g(t)');
grid on;
```

Figure 14: MATLAB code for simulating closed-loop step response and performance characteristics of nonlinear system (code for displaying performance information not included)

A-posteriori Analysis & Results

Closed-Loop Step Response of Linearized System

The following graph shows the step responses of the closed-loop system, both with and without simplified gains:

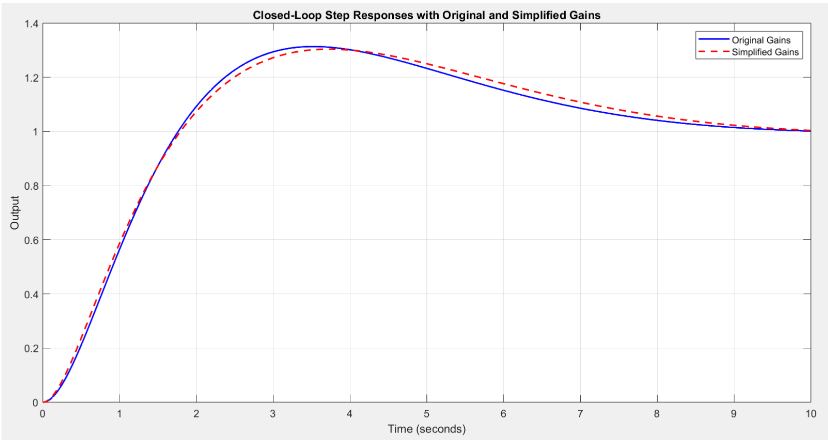

Figure 15: Closed-loop Step Response with and without Simplified Gains (generated by code from figure 13)

Furthermore, here are the performance metrics of the closed-loop system with simplified gains:

- **Steady-state error: 1.0004** (approximately 1, meets specification)
- **Settling Time: 9.11 seconds** (meets specification)
- **PO: 30.40%** (less than 41.90%, meets specification)
- **System is stable** (meets specification)
  To determine that the closed-loop system is stable, the eigenvalues of the augmented system matrix had to have negative real parts:

$$\begin{bmatrix} A - BK_{simplified} & BK_{simplified} \\ 0 & A - L_{simplified}C \end{bmatrix} \tag{24}$$

  Since the eigenvalues of the above matrix are:

$$-3.1479,$$

$$-4.2636,$$

$$-0.4261 \pm 0.3690j,$$
$$-9.5282 \pm 7.9345j$$

therefore, the closed-loop system is stable since all of the eigenvalues have negative real parts to them (all on left side of complex plane). **Because of this, the linearized closed-loop system meets the specifications**.

Failsafe Investigation

Using equation (22) and scaling the respective gains for K and L accordingly, here are the results for the failsafe investigation:

- Gains K and L both set to 50% of their nominal values

  Eigenvalues: $-9.4026, -11.5722, -0.2987 \pm 1.1373j, -4.0439 \pm 3.6123j$

  Since all of the eigenvalues have negative real parts, this gain combination is stable.

- Gains K and L both set to 200% of their nominal values

  Eigenvalues: $8.8576, -3.5428, -0.4288 \pm 1.5533j, -13.5486 \pm 13.3947j$

  Since one of the eigenvalues ($s = 8.8576$) has a positive real part, this gain combination is not stable.

- Gain K set to 200% and gain L set to 50% of nominal values

  Eigenvalues: $8.8576, -11.5722, -0.4288 \pm 1.5533j, -4.0439 \pm 3.6123j$

  Since one of the eigenvalues ($s = 8.8576$) has a positive real part, this gain combination is not stable.

- Gain L set to 200% and gain K set to 50% of nominal values

  Eigenvalues: $-9.4026, -3.5428, -0.2987 \pm 1.1373j, -13.5486 \pm 13.3947j$

  Since all of the eigenvalues have negative real parts, this gain combination is stable.

Of the four gain combinations tested, only two of them were stable, which was when K was at 50% of its nominal value. Whenever the gain of K failed high at 200% of its nominal value, the closed-loop system became unstable. Because the system was not failsafe across two of the four gain combinations tested due to becoming unstable at the extremes (in this case when the gain K went to 50% of its nominal value), **the linearized system is not failsafe**. Furthermore, there is no need to test for more gain combinations because the instability observed at the tested extremes indicates a fundamental limitation in the system's design. Additional tests are unlikely to yield different results and would not change the conclusion that the system is not failsafe.

Performance of Actual Nonlinear Closed-loop System

The following graph shows the step response of the nonlinear closed-loop system:

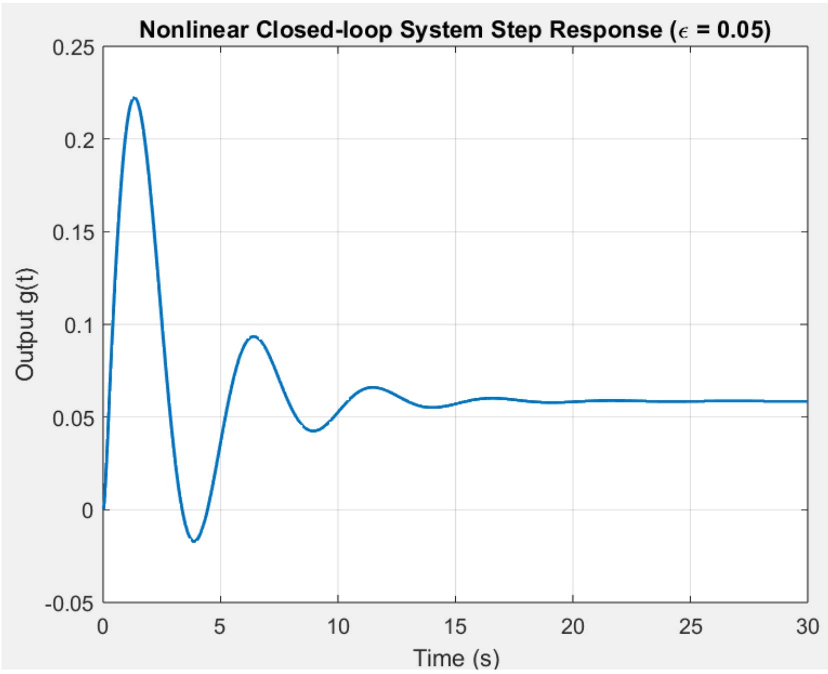

Figure 16: Step Response of Nonlinear Closed-loop System (generated by code from figure 14)

The performance characteristics for the nonlinear system are as follows:

- The closed-loop system is stable (converges to a final steady-state value of 0.0585) $\Rightarrow$ **the stability requirement is met**
- Steady-state error: 0.0415 $\Rightarrow$ there is nonzero steady-state error, so **the steady-state error requirement is not met**

- 2% settling time: 0.2569 seconds ⇒ since this is in the acceptable $t_s(2\%)$ range (see page 23), **the 2% settling time requirement is met**
- PO%: 279.62% ⇒since this is greater than the maximum allowed percent overshoot of 41.9%, **the percent overshoot requirement is not met**
- Peak Time: 1.3206 seconds

While the nonlinear closed-loop system met some of the specifications, it failed to meet all of them. This means that to meet the stated design specifications, the system must be linearized about the operating point, in this case by employing a controller and observer via linear state feedback.