Table of Contents

Hw2

Author: Thomas Dunnington Modified: 2/8/2025

```
close all; clear; clc;
```

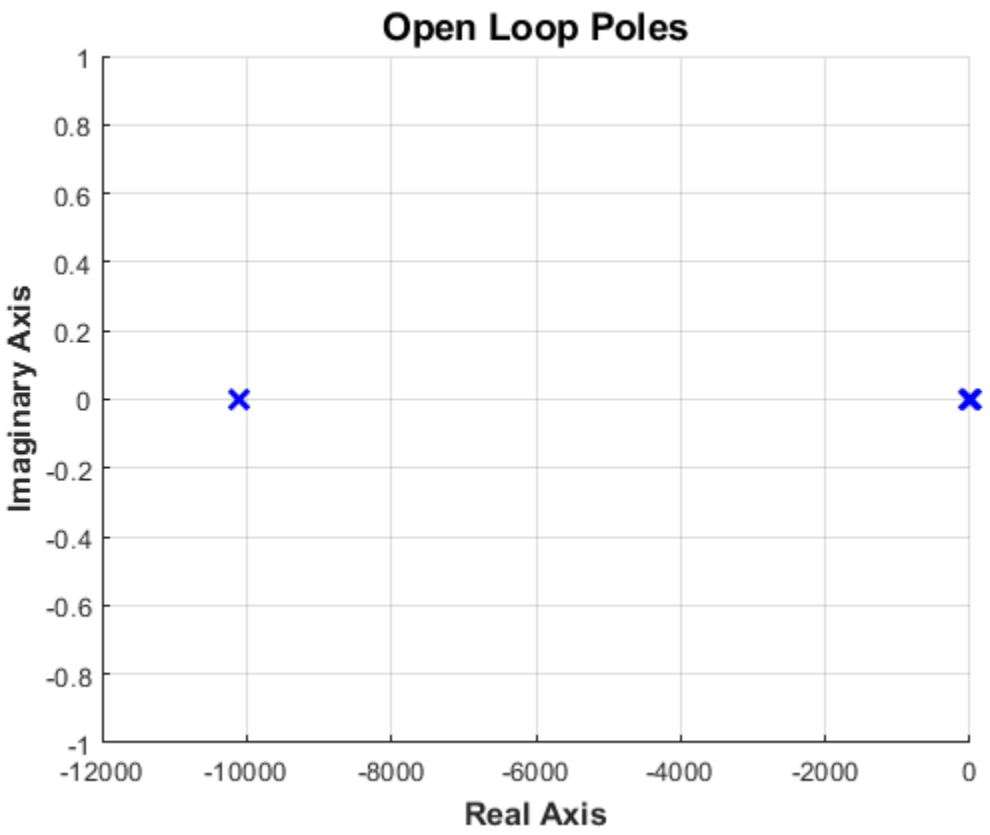
Constants

```
Rm = 19.2;                    % ohm
Lm = 1.9 / 1000;              % H
Kt = 40.1 / 1000;             % Nm/A
Jm = 12.5 / (1000 * 100^2); % kgm^2
Kb = 1/238 * 1/(2*pi) * 60; % V/(rad/s)
N = 10 * 120 / 36;

% Constants calculated from Hw1
Jl = 0.0154 + 0.0013;
Jeq = 0.0179;
Ks = 0.7990;
```

Problem 1

Vp to ThetaL transfer function

```
num_thetaVp = [-1];
den_thetaVp = [Jeq*Lm / (Kt*N), Jeq*Rm / (Kt*N), Kb*N, 0];

% Find the roots
poles = roots(den_thetaVp);

% Plot in the complex plane
figure();
scatter(real(poles), imag(poles), 100, 'b', 'Marker', 'x', 'linewidth', 2) %
Larger, filled blue markers
xlabel('Real Axis', 'FontSize', 12, 'FontWeight', 'bold')
ylabel('Imaginary Axis', 'FontSize', 12, 'FontWeight', 'bold')
title('Open Loop Poles', 'FontSize', 14, 'FontWeight', 'bold')
grid on
```

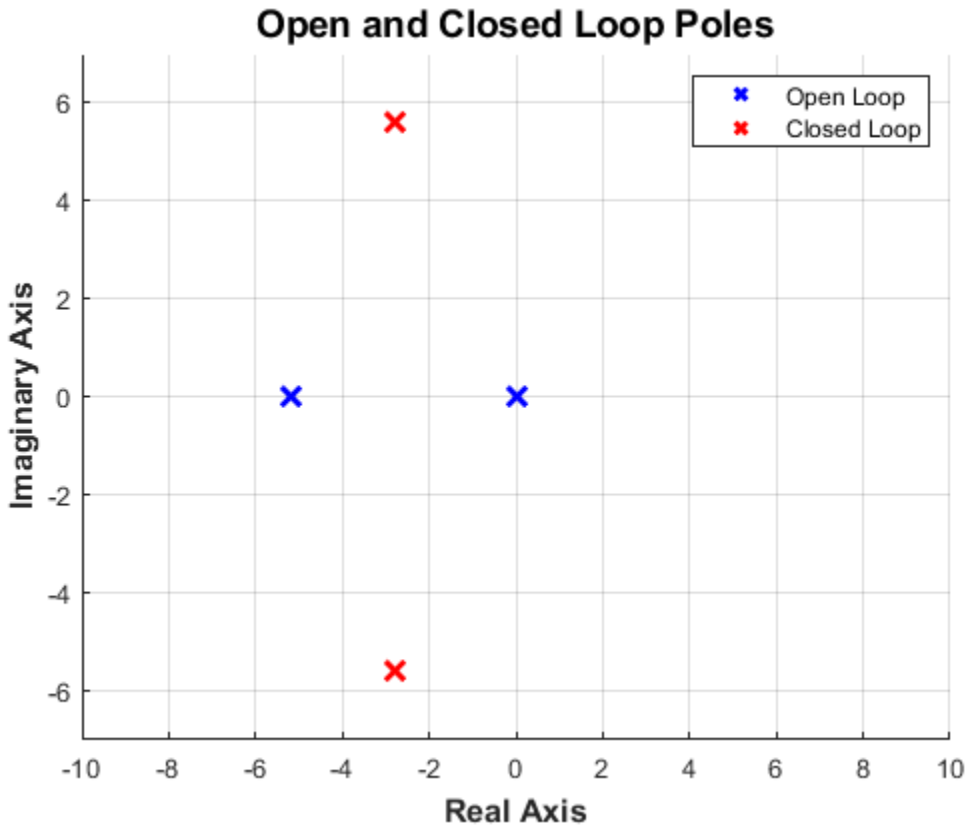


Problem 2

Transfer function theta R and theta L

```
Gp = -10;
Gd = -0.1;

% Define closed loop transfer function
num_closed = @(Gp, Gd)[Gd Gp];
den_closed = @(Gp, Gd)[-Jeq*Lm / (Kt*N), -Jeq*Rm / (Kt*N), Gd - Kb*N, Gp];

% Plug in standard gain values
num_closed_1 = num_closed(Gp, Gd);
den_closed_1 = den_closed(Gp, Gd);

% Find the poles
poles_1 = roots(den_closed_1);

% Plot in the complex plane
figure();
scatter(real(poles), imag(poles), 100, 'b', 'Marker', 'x', 'linewidth', 2)
hold on
scatter(real(poles_1), imag(poles_1), 100, 'r', 'Marker', 'x', 'linewidth',
2)
xlabel('Real Axis', 'FontSize', 12, 'FontWeight', 'bold')
ylabel('Imaginary Axis', 'FontSize', 12, 'FontWeight', 'bold')
```

```
title('Open and Closed Loop Poles', 'FontSize', 14, 'FontWeight', 'bold')
grid on
legend('Open Loop', 'Closed Loop')
xlim([-10 10])
ylim([-7 7])
```

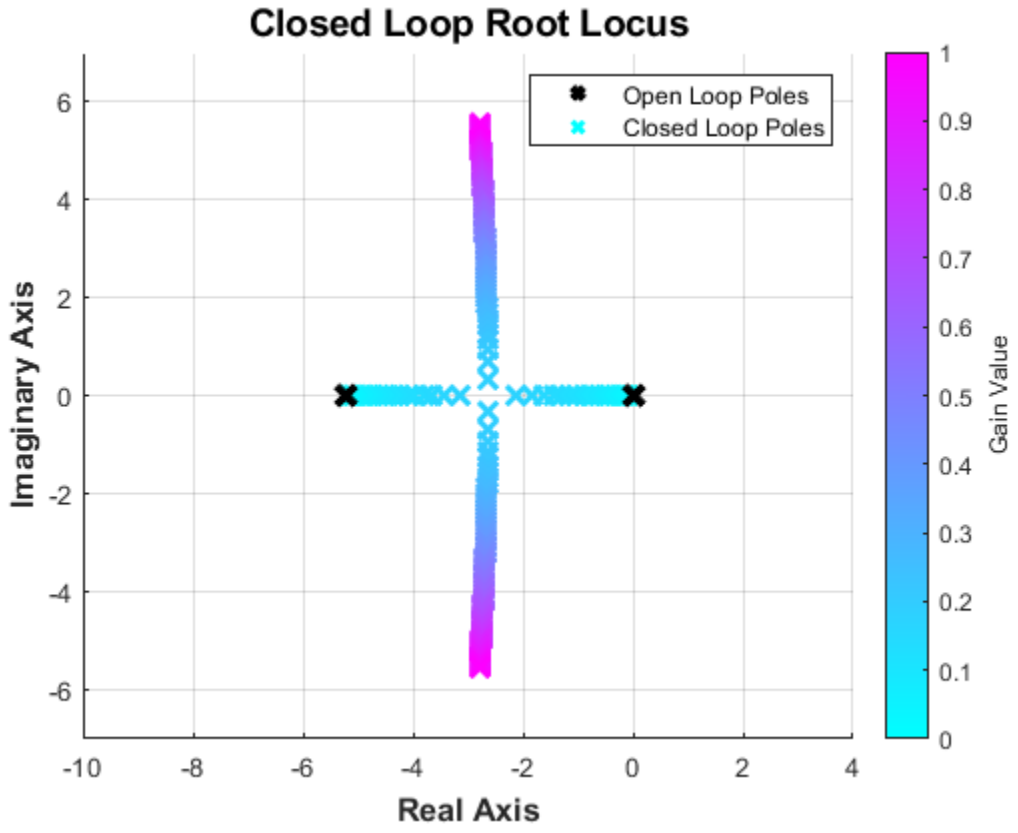


Problem 3

Gain factor for root locus

```
g = linspace(0, 1, 100);

% Find poles
poles_root_locus = zeros(length(g), length(poles_1));
for i = 1:length(g)
    den_closed_temp = den_closed(g(i)*Gp, g(i)*Gd);
    poles_root_locus(i,:) = roots(den_closed_temp);
end

% Plot the root locus
figure();
colormap(cool);
closed_plot = scatter(real(poles_root_locus(1,:)),
imag(poles_root_locus(1,:)), 100, g(1).*ones(3,1), 'Marker', 'x',
'linewidth', 2);
hold on
```

```
for i = 2:length(poles_root_locus(2:end,1))
    scatter(real(poles_root_locus(i,:)), imag(poles_root_locus(i,:)), 100,
g(i).*ones(3,1), 'Marker', 'x', 'linewidth', 2)
end
open_plot = scatter(real(poles), imag(poles), 100, 'k', 'Marker', 'x',
'linewidth', 3);
xlabel('Real Axis', 'FontSize', 12, 'FontWeight', 'bold')
ylabel('Imaginary Axis', 'FontSize', 12, 'FontWeight', 'bold')
title('Closed Loop Root Locus', 'FontSize', 14, 'FontWeight', 'bold')
legend([open_plot, closed_plot], 'Open Loop Poles', 'Closed Loop Poles')
grid on
colorbar;
clim([0 1]); % Set limits to actual gain range
ylabel(colorbar, 'Gain Value');
xlim([-10 4])
ylim([-7 7])
```

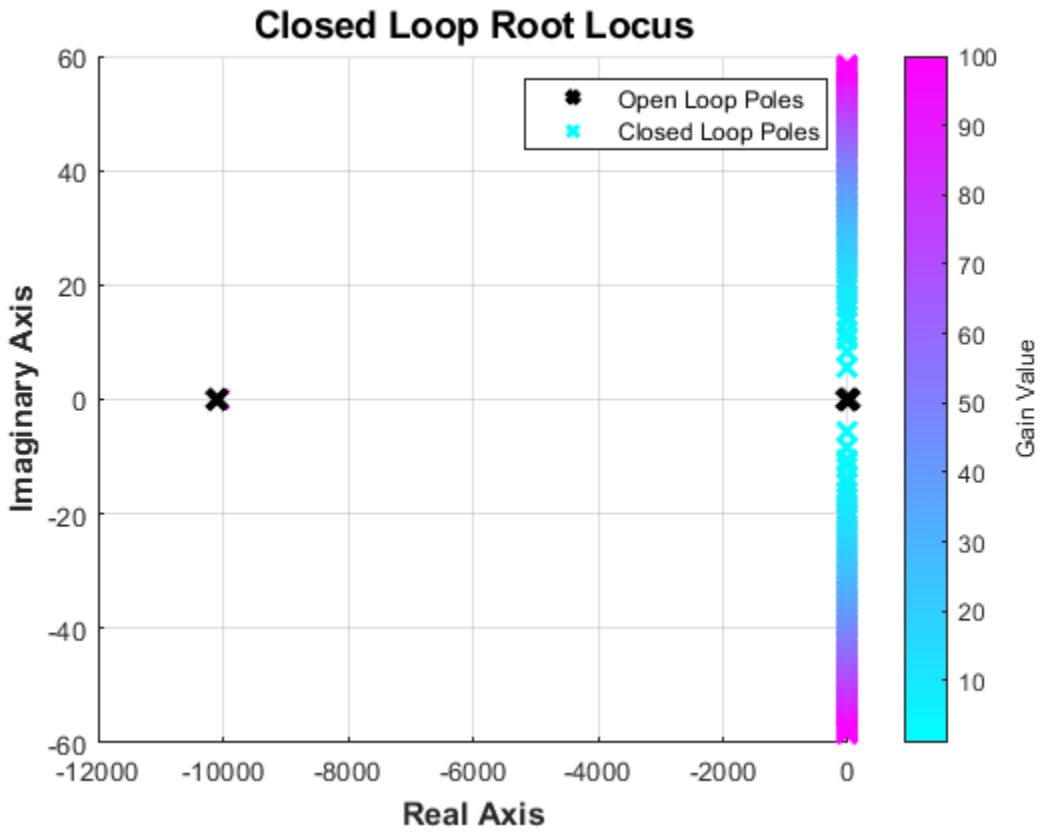


Problem 4

Gain factor for root locus

```
g = linspace(1, 100, 100);

% Find poles
poles_root_locus = zeros(length(g), length(poles_1));
for i = 1:length(g)
```

```
    den_closed_temp = den_closed(g(i)*Gp, g(i)*Gd);
    poles_root_locus(i,:) = roots(den_closed_temp);
end

% Plot the root locus
figure();
colormap(cool);
closed_plot = scatter(real(poles_root_locus(1,:)),
imag(poles_root_locus(1,:)), 100, g(1).*ones(3,1), 'Marker', 'x',
'linewidth', 2);
hold on
for i = 2:length(poles_root_locus(2:end,1))
    scatter(real(poles_root_locus(i,:)), imag(poles_root_locus(i,:)), 100,
g(i).*ones(3,1), 'Marker', 'x', 'linewidth', 2)
end
open_plot = scatter(real(poles), imag(poles), 100, 'k', 'Marker', 'x',
'linewidth', 3);
xlabel('Real Axis', 'FontSize', 12, 'FontWeight', 'bold')
ylabel('Imaginary Axis', 'FontSize', 12, 'FontWeight', 'bold')
title('Closed Loop Root Locus', 'FontSize', 14, 'FontWeight', 'bold')
legend([open_plot, closed_plot], 'Open Loop Poles', 'Closed Loop Poles')
grid on
colorbar;
clim([1 100]); % Set limits to actual gain range
ylabel(colorbar, 'Gain Value');
% xlim([-65 65])
% ylim([-65 65])
```

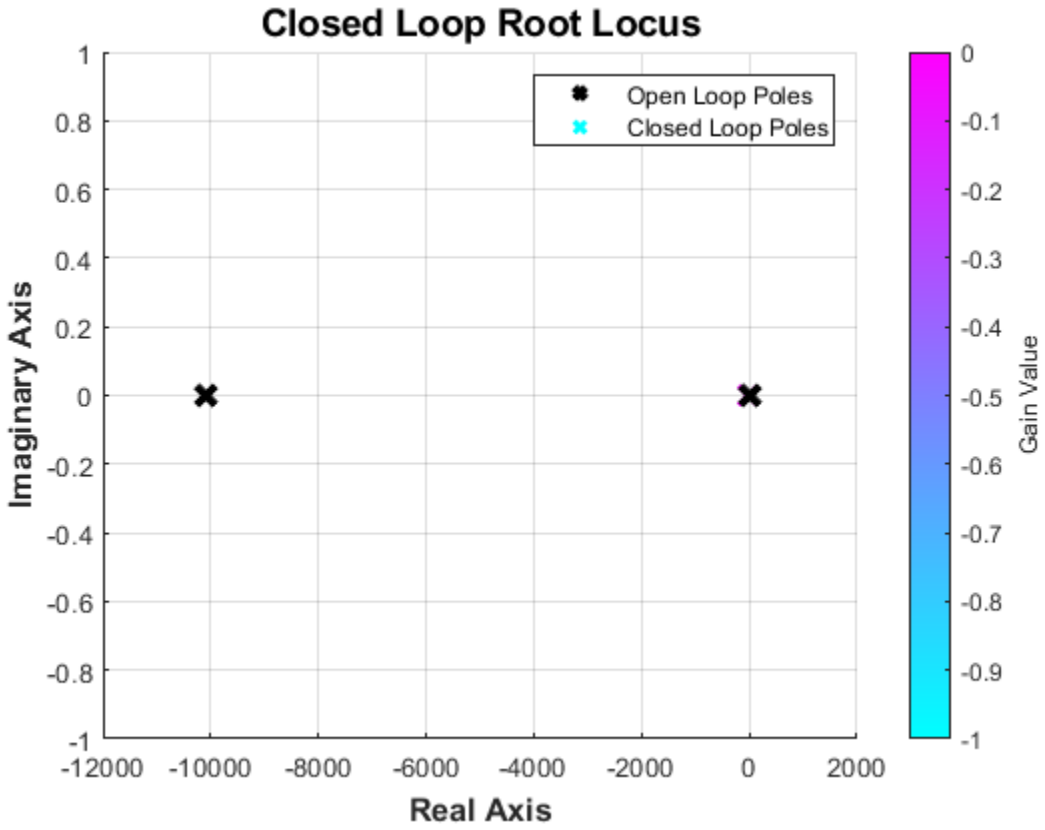


Problem 5

Gain factor for root locus

```
g = linspace(-1, 0, 100);

% Find poles
poles_root_locus = zeros(length(g), length(poles_1));
for i = 1:length(g)
    den_closed_temp = den_closed(g(i)*Gp, g(i)*Gd);
    poles_root_locus(i,:) = roots(den_closed_temp);
end

% Plot the root locus
figure();
colormap(cool);
closed_plot = scatter(real(poles_root_locus(1,:)),
imag(poles_root_locus(1,:)), 100, g(1).*ones(3,1), 'Marker', 'x',
'linewidth', 2);
hold on
for i = 2:length(poles_root_locus(2:end,1))
    scatter(real(poles_root_locus(i,:)), imag(poles_root_locus(i,:)), 100,
g(i).*ones(3,1), 'Marker', 'x', 'linewidth', 2)
end
open_plot = scatter(real(poles), imag(poles), 100, 'k', 'Marker', 'x',
'linewidth', 3);
```

```
xlabel('Real Axis', 'FontSize', 12, 'FontWeight', 'bold')
ylabel('Imaginary Axis', 'FontSize', 12, 'FontWeight', 'bold')
title('Closed Loop Root Locus', 'FontSize', 14, 'FontWeight', 'bold')
legend([open_plot, closed_plot], 'Open Loop Poles', 'Closed Loop Poles')
grid on
colorbar;
clim([-1 0]); % Set limits to actual gain range
ylabel(colorbar, 'Gain Value');
% xlim([-65 65])
% ylim([-65 65])
```

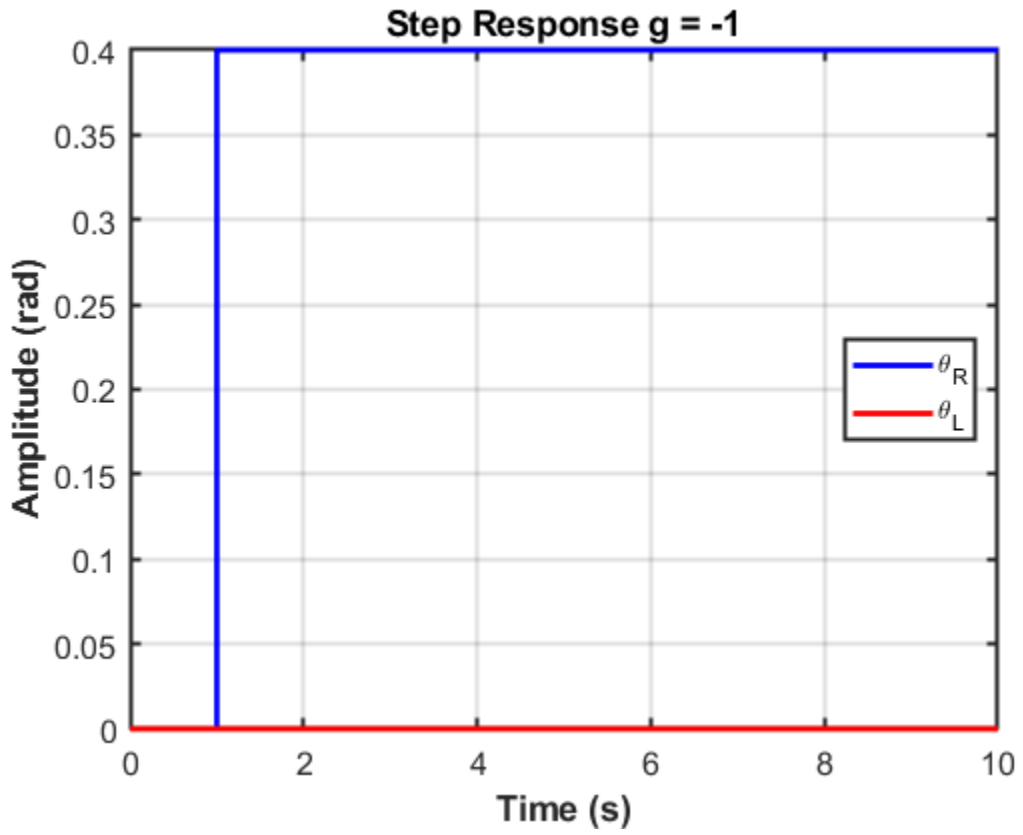


Problem 6

Simulate the closed loop response

```
g = 0;
Gp = g * Gp;
Gd = g * Gd;

output = sim('thetaRef_thetaL.slx');
t = output.t;
thetaR = output.thetaR;
thetaL = output.thetaL;
Vp = output.Vp;
```

```
% System Response
figure();
plot(t, thetaR, 'b-', 'LineWidth', 2); % Blue solid line, thicker
hold on;
plot(t, thetaL, 'r', 'LineWidth', 2); % Red dashed line, thicker

xlabel('Time (s)', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('Amplitude (rad)', 'FontSize', 12, 'FontWeight', 'bold');
title('Step Response g = -1', 'FontSize', 14, 'FontWeight', 'bold');

legend('\theta_R', '\theta_L', 'Location', 'best', 'FontSize', 10);
grid on; % Enable grid for better readability
xlim([min(t), max(t)]); % Adjust x-axis limits based on data
ylim auto; % Auto-scale y-axis
set(gca, 'FontSize', 12, 'LineWidth', 1.5); % Improve axis appearance
```



Published with MATLAB® R2024a