EEE575

ASSIGNMENT 4

Q1

The paper by Marcel Carvalho and Minhoto Teixeira introduces a comprehensive method to simplify the traditional Ogata lead-lag compensatory design by developing direct analytical expressions. This approach addresses a fundamental limitation of the classical root locus technique, which typically relies on an iterative trial-and-error process to determine suitable compensatory parameters

The key contribution is the derivation of closed-form mathematical expressions that allow for the direct calculation of the compensatory parameters based on desired closed-loop specifications, such as damping ratio or settling time. Unlike the iterative graphical method, these expressions enable designers to compute the parameters algebraically, eliminating the need for repetitive pole-zero placement and root locus sketching.

Building on Ogata's foundational work, this method bypasses the graphical construction steps. The authors demonstrate that by correctly formulating design constraints such as mathematical relationships, the parameters can be solved directly.

A significant advantage of this direct approach is its suitability for computer-aided design environments. The expressions can be easily implemented in software tools, enabling rapid design iteration and optimization. This is particularly valuable in education, allowing students to focus on core control concepts rather than tedious graphical procedures.

The method is validated through design examples, showing comparable performance to the traditional approach but with a substantial reduction in design time. This work effectively bridges classical control theory and modern computational tools, making advanced compensation design more accessible while preserving the physical intuition of the root locus method.

Q2

$$G(s) = \frac{5}{s(s+2)}$$

$$H(s) = 1 + \alpha s$$

$$\begin{aligned}
T(s) &= \frac{G(s)}{1 + G(s)H(s)} \\
&= \frac{\frac{5}{s(s+2)}}{1 + \frac{5(1+\alpha s)}{s(s+2)}} \\
&= \frac{5}{s(s+2) + 5(1+\alpha s)} \\
&= \frac{5}{s^2 + (2 + 5\alpha)\ s + 5}
\end{aligned}$$

$$T(s) = \frac{5}{s^2 + (2 + 5\alpha)\ s + 5}$$

Characteristic equation

$$s^2 + (2 + 5\alpha)\ s + 5$$

$$1 + KL(s) = 0$$

$$(s^2 + 2s + 5) + \alpha(5s) = 0$$

$$1 + \alpha \frac{5s}{s^2 + 2s + 5} = 0$$

$$L(s) = \frac{b(s)}{a(s)} = \frac{5s}{s^2 + 2s + 5}$$

$$\begin{aligned} a(s) &= s^2 + 2s + 5 \\ b(s) &= 5s \end{aligned}$$

$$a(s) = s^2 + 2s + 5 = 0$$

$$s = -1 \pm 2i$$

$$b(s) = 5s = 0$$

$$s = 0$$

$$\begin{aligned} L(s) &= \frac{5s}{s^2 + 2s + 5} \\ a(s) &= s^2 + 2s + 5 \\ b(s) &= 5s \\ pole\colon\ s &= -1 \pm 2j \\ zero\colon\ s &= 0 \end{aligned}$$

n = 2 (number of poles)

m = 1 (number of zeros)

Root locus plotting

Starting point (K=0): Two branches start at the pole -1±2j

End point (K=∞): One branch ends at zero 0, and one branch ends at ∞

The root locus on the real axis is in a region with an odd number of poles and zeros to its right:

The zero is at s = 0, and the pole is at s = -1 ± 2j (not on the real axis).

Interval (-∞, 0): 1 zero to the right (odd number) → root locus exists.

Interval (0, +∞): 0 poles and zeros to the right (even number) → no root locus exists.

For n-m = 2-1 = 1, 1 asymptote

$$\theta_{\mathrm{k}} = \frac{180° + 360°k}{n-m} = \frac{180° + 360°k}{1}$$

$$k = 0, \theta_0 = 180°$$

$$\sigma_{\mathrm{a}} = \frac{\sum pole - \sum zero}{n-m} = \frac{(-1+2j) + (-1-2j) - 0}{2-1} = \frac{-2}{1} = -2$$

Asymptote: A straight line passing through the point (-2, 0) with an angle of 180° (i.e., the negative direction of the real axis)

$$z_1 - s_2 = 0 - (-1-2j) = 1 + 2j$$

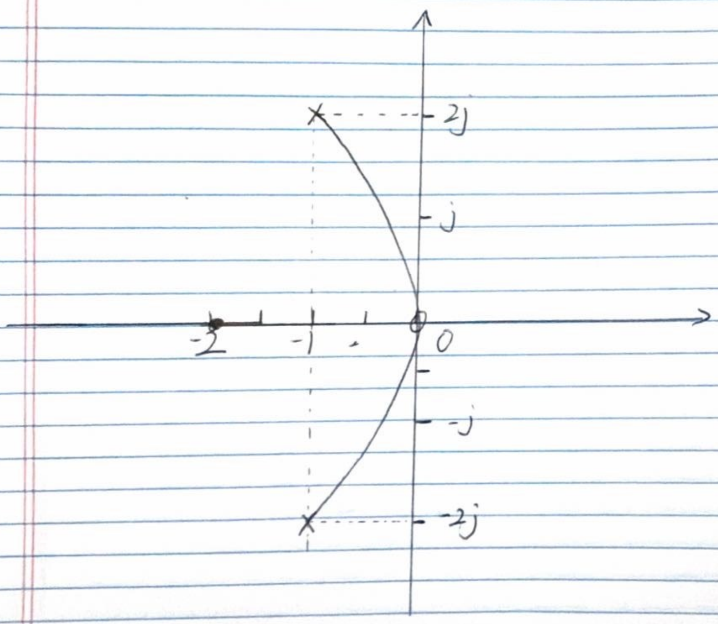
$$\angle(z^1 - s^2) = \angle(1+2j) = \arctan\left(\frac{2}{1}\right) = 63.43°$$

$$s^1 - s^2 = (-1+2j) - (-1-2j) = 4j$$

$$\angle(s_1 - s_2) = \angle(4j) = 90°$$

$$\theta_d eparture = 63.43° - 90° + 180°$$

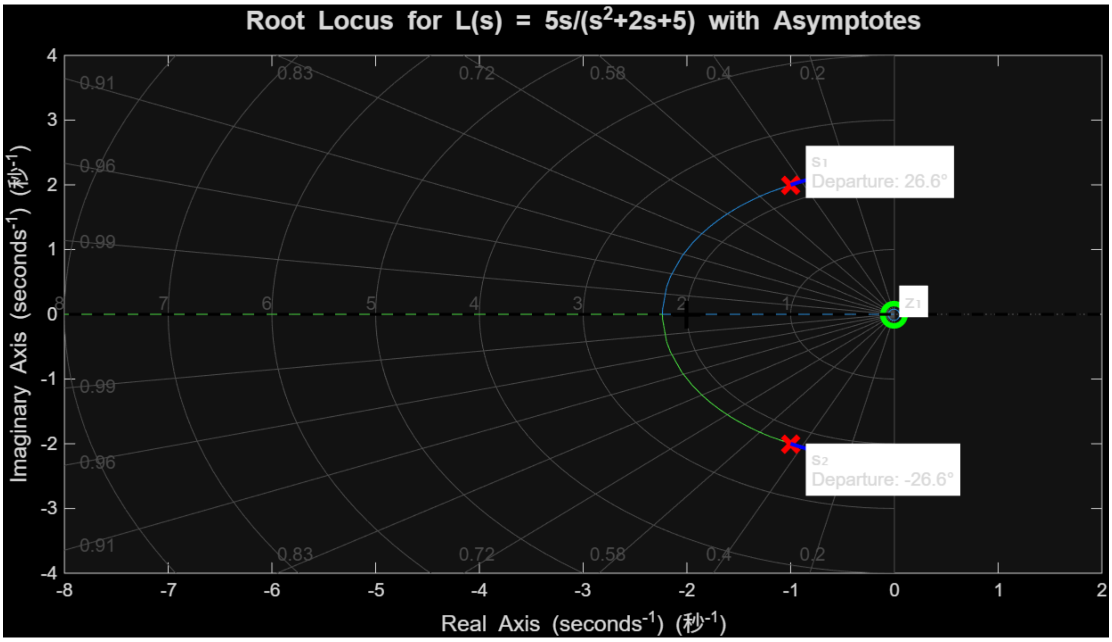
$$\theta_d eparture = 153.43°$$

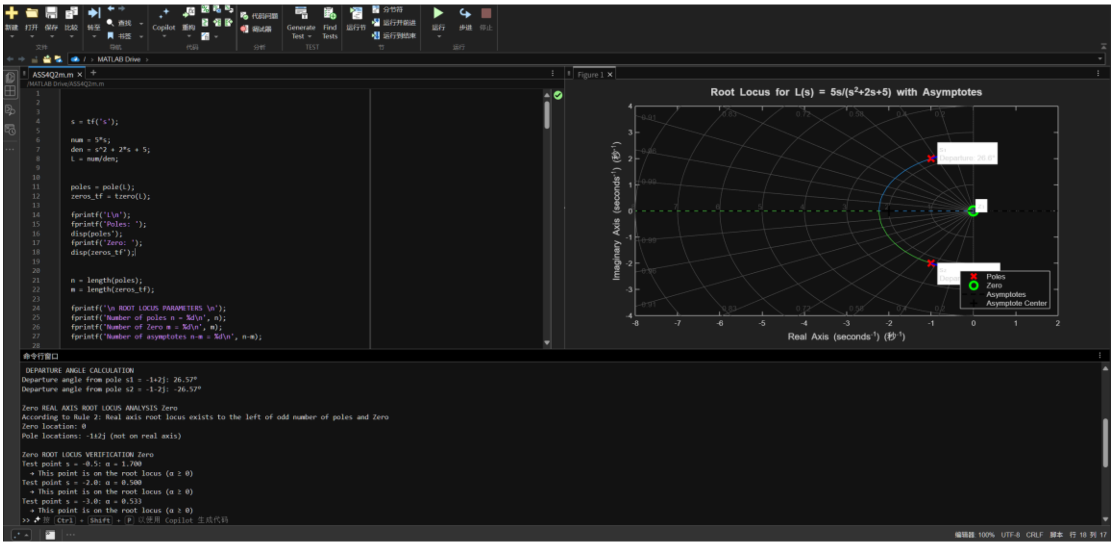


Asymptote: A straight line passing through the point (-2,0) with an angle of 180°

```
>> ASS4Q2m
L
Poles:   -1.0000 - 2.0000i  -1.0000 + 2.0000i

Zero:      0


 ROOT LOCUS PARAMETERS
Number of poles n = 2
Number of Zero m = 1
Number of asymptotes n-m = 1

 ASYMPTOTE CALCULATION
Asymptote center: σ = -2.00
Asymptote angles: 180.0°
```

```
 DEPARTURE ANGLE CALCULATION
Departure angle from pole s1 = -1+2j: 26.57°
Departure angle from pole s2 = -1-2j: -26.57°

Zero REAL AXIS ROOT LOCUS ANALYSIS Zero
According to Rule 2: Real axis root locus exists to the left of odd
number of poles and Zero
Zero location: 0
Pole locations: -1±2j (not on real axis)

Zero ROOT LOCUS VERIFICATION Zero
Test point s = -0.5: α = 1.700
  → This point is on the root locus (α ≥ 0)
Test point s = -2.0: α = 0.500
  → This point is on the root locus (α ≥ 0)
Test point s = -3.0: α = 0.533
  → This point is on the root locus (α ≥ 0)
```





Q3

$$G(s) = \frac{1}{(s+2)(s+3)}$$

$$D_c(s) = K\frac{s+a}{s+b}$$

$$L(s) = D_c(s)G(s) = K\frac{(s+a)}{(s+b)(s+2)(s+3)}$$

$$1 + L(s) = 0$$

$$1 + K\frac{s+a}{(s+b)(s+2)(s+3)} = 0$$

$$(s+b)(s+2)(s+3) + K(s+a) = 0$$

$$s = -1 \pm j$$

So, let's use -1-j

$$(s+b)(s+2)(s+3) + K(s+a) = 0$$

$$(b-1-j)(-1-j+2)(-1-j+3) + K(-1-j+a) = 0$$

$$(1-3i)(b-(1+i)) + K(a-(1+i)) = 0$$

$$(1-3j)(b-(1+j))+K(a-(1+j))=0$$

$$(b-4)+(2-3b)j+(a-1)K-jk=0$$

$$\begin{aligned} b-4+K(a-1) &= 0 \\ (2-3b)-K &= 0 \end{aligned}$$

$$k=2-3b$$

$$a=\frac{6-4b}{2-3b}$$

Let $b=0$,

$$K=2, a=3$$

$$D_c(s)=K\frac{s+a}{s+b}=\frac{2(s+3)}{s}=\frac{2s+6}{s}$$

$$D_c(s)=\frac{2s+6}{s}$$

$$G(s)=\frac{1}{(s+2)(s+3)}$$

$$L(s)=Dc(s)G(s)=\frac{2s+6}{s(s+2)(s+3)}$$

$$L(s)=D_c(s)G(s)=2\frac{s+3}{s(s+2)(s+3)}=\frac{2}{s(s+2)}$$

$$D_c(s)=\frac{2s+6}{s}$$

$$1+L(s)=0$$

$$1+2L(s)=0$$

$$
\begin{aligned}
1+\frac{2}{s(s+2)} &= 0 \\
\frac{s(s+2)+2}{s(s+2)} &= 0 \\
\frac{s^2+2s+2}{s(s+2)} &= 0 \\
s^2+2s+2 &= 0
\end{aligned}
$$

$$s^2+2s+2 = 0$$

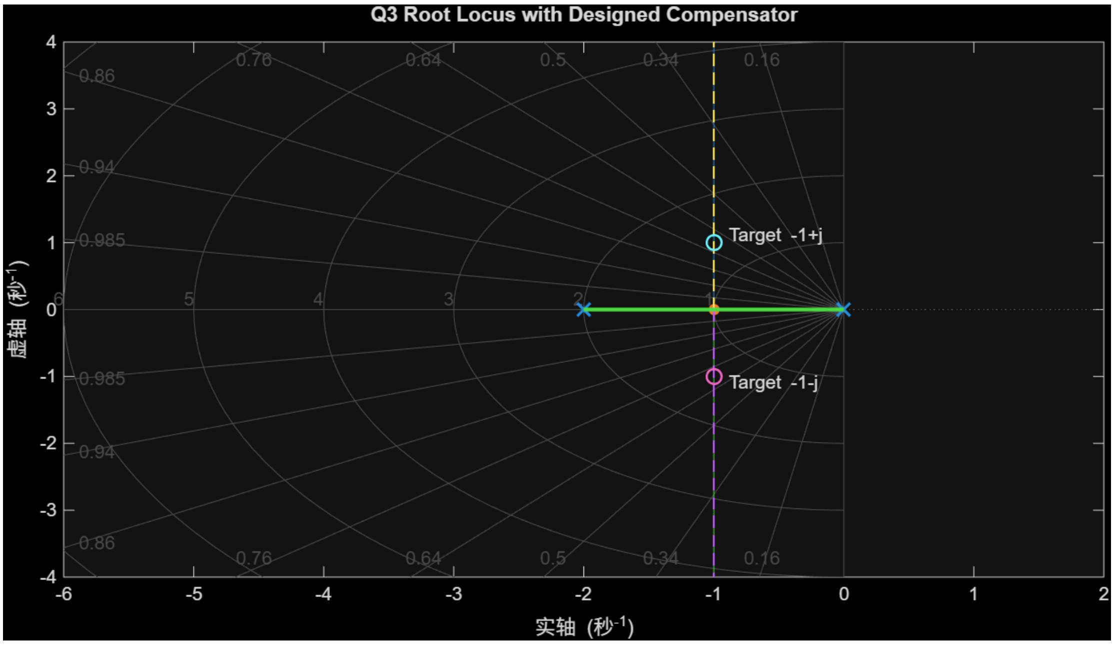
$$s = 1 \pm j$$

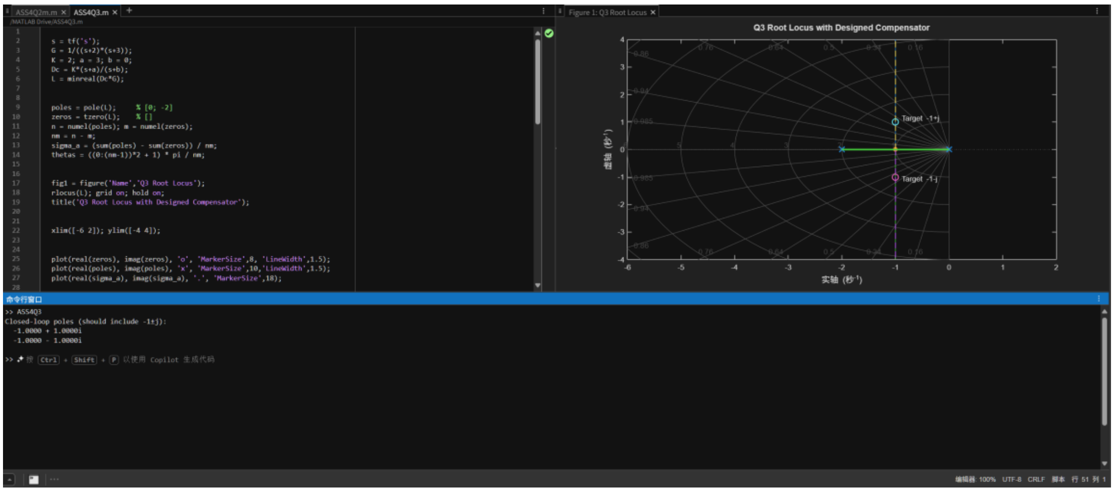
$$k = 2, a = 3, b = 0$$

$$D_c(s) = 2\frac{s+3}{s}$$

This is a **lag compensator** because it approximates PI control (pole at origin, zero at -3), improving steady-state accuracy.

```
>> ASS4Q3
Closed-loop poles (should include -1±j):
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
```





Code

Q2

```
s = tf('s');

num = 5*s;
den = s^2 + 2*s + 5;
L = num/den;
```

```matlab
poles = pole(L);
zeros_tf = tzero(L);

fprintf('L\n');
fprintf('Poles: ');
disp(poles');
fprintf('Zero: ');
disp(zeros_tf');


n = length(poles);
m = length(zeros_tf);

fprintf('\n ROOT LOCUS PARAMETERS \n');
fprintf('Number of poles n = %d\n', n);
fprintf('Number of Zero m = %d\n', m);
fprintf('Number of asymptotes n-m = %d\n', n-m);


if n > m

    sigma_a = (sum(real(poles)) - sum(real(zeros_tf))) / (n - m);


    asymptote_angles = zeros(1, n-m);
    for k = 0:(n-m-1)
        asymptote_angles(k+1) = (180 + 360*k) / (n-m);
    end

    fprintf('\n ASYMPTOTE CALCULATION \n');
    fprintf('Asymptote center: σ = %.2f\n', sigma_a);
    fprintf('Asymptote angles: ');
    for i = 1:length(asymptote_angles)
        fprintf('%.1f° ', asymptote_angles(i));
    end
    fprintf('\n');
end


s1 = -1 + 2j;
s2 = -1 - 2j;
zero = 0;


angle_to_zero_from_s1 = angle(zero - s1) * 180/pi;
```

```
angle_from_s2_to_s1 = angle(s1 - s2) * 180/pi;
departure_s1 = angle_to_zero_from_s1 - angle_from_s2_to_s1 + 180;


angle_to_zero_from_s2 = angle(zero - s2) * 180/pi;
angle_from_s1_to_s2 = angle(s2 - s1) * 180/pi;
departure_s2 = angle_to_zero_from_s2 - angle_from_s1_to_s2 + 180;


departure_s1 = mod(departure_s1 + 180, 360) - 180;
departure_s2 = mod(departure_s2 + 180, 360) - 180;

fprintf('\n DEPARTURE ANGLE CALCULATION \n');
fprintf('Departure angle from pole s1 = -1+2j: %.2f°\n', departure_s1);
fprintf('Departure angle from pole s2 = -1-2j: %.2f°\n', departure_s2);


figure('Position', [100, 100, 1200, 800]);


rlocus(L);
hold on;
grid on;


plot(real(poles), imag(poles), 'rx', 'MarkerSize', 12, 'LineWidth', 3);
plot(real(zeros_tf), imag(zeros_tf), 'go', 'MarkerSize', 12,
'LineWidth', 3);


if n > m

    asymptote_length = 10;

    for i = 1:length(asymptote_angles)
        angle_rad = asymptote_angles(i) * pi / 180;


        x_end = sigma_a + asymptote_length * cos(angle_rad);
        y_end = 0 + asymptote_length * sin(angle_rad);

        x_start = sigma_a - asymptote_length * cos(angle_rad);
        y_start = 0 - asymptote_length * sin(angle_rad);


        plot([x_start, x_end], [y_start, y_end], 'k--', ...
```

```
            'LineWidth', 1.5, 'DisplayName', sprintf('Asymptote %.1f°',
asymptote_angles(i)));
    end


    plot(sigma_a, 0, 'k+', 'MarkerSize', 15, 'LineWidth', 2, ...
         'DisplayName', 'Asymptote Center');
end


arrow_length = 1;


departure_rad_s1 = departure_s1 * pi / 180;
x_arrow_s1 = real(s1) + arrow_length * cos(departure_rad_s1);
y_arrow_s1 = imag(s1) + arrow_length * sin(departure_rad_s1);
quiver(real(s1), imag(s1), x_arrow_s1 - real(s1), y_arrow_s1 -
imag(s1), ...
       0, 'b', 'LineWidth', 2, 'MaxHeadSize', 0.3, 'DisplayName',
'Departure Angle');


departure_rad_s2 = departure_s2 * pi / 180;
x_arrow_s2 = real(s2) + arrow_length * cos(departure_rad_s2);
y_arrow_s2 = imag(s2) + arrow_length * sin(departure_rad_s2);
quiver(real(s2), imag(s2), x_arrow_s2 - real(s2), y_arrow_s2 -
imag(s2), ...
       0, 'b', 'LineWidth', 2, 'MaxHeadSize', 0.3);


text(real(s1) + 0.2, imag(s1) + 0.2, sprintf('s₁\nDeparture: %.1f°',
departure_s1), ...
     'FontSize', 10, 'BackgroundColor', 'white');
text(real(s2) + 0.2, imag(s2) - 0.4, sprintf('s₂\nDeparture: %.1f°',
departure_s2), ...
     'FontSize', 10, 'BackgroundColor', 'white');
text(real(zeros_tf) + 0.1, imag(zeros_tf) + 0.2, 'z₁', ...
     'FontSize', 10, 'BackgroundColor', 'white');


title('Root Locus for L(s) = 5s/(s^2+2s+5) with Asymptotes',
'FontSize', 14);
xlabel('Real Axis (seconds^{-1})', 'FontSize', 12);
ylabel('Imaginary Axis (seconds^{-1})', 'FontSize', 12);
```

```
h_legend = [];
labels_legend = {};


h_rlocus = findobj(gca, 'Type', 'line', 'Color', 'b');
if ~isempty(h_rlocus)
    h_legend(end+1) = h_rlocus(1);
    labels_legend{end+1} = 'Root Locus';
end


h_pole = findobj(gca, 'Marker', 'x', 'Color', 'r');
if ~isempty(h_pole)
    h_legend(end+1) = h_pole(1);
    labels_legend{end+1} = 'Poles';
end


h_zero = findobj(gca, 'Marker', 'o', 'Color', 'g');
if ~isempty(h_zero)
    h_legend(end+1) = h_zero(1);
    labels_legend{end+1} = 'Zero';
end


h_asym = findobj(gca, 'LineStyle', '--', 'Color', 'k');
if ~isempty(h_asym)
    h_legend(end+1) = h_asym(1);
    labels_legend{end+1} = 'Asymptotes';
end


h_center = findobj(gca, 'Marker', '+', 'Color', 'k');
if ~isempty(h_center)
    h_legend(end+1) = h_center(1);
    labels_legend{end+1} = 'Asymptote Center';
end

legend(h_legend, labels_legend, 'Location', 'best', 'FontSize', 10);


xlim([-8, 2]);
ylim([-4, 4]);

hold off;
```

```
fprintf('\nZero REAL AXIS ROOT LOCUS ANALYSIS Zero\n');
fprintf('According to Rule 2: Real axis root locus exists to the left
of odd number of poles and Zero\n');
fprintf('Zero location: 0\n');
fprintf('Pole locations: -1±2j (not on real axis)\n');



fprintf('\nZero ROOT LOCUS VERIFICATION Zero\n');
test_points = [-0.5, -2, -3];
for i = 1:length(test_points)
    s_test = test_points(i);
    L_value = 5*s_test / (s_test^2 + 2*s_test + 5);
    alpha_required = -1 / L_value;

    fprintf('Test point s = %.1f: α = %.3f\n', s_test, alpha_required);

    if alpha_required >= 0
        fprintf('  → This point is on the root locus (α ≥ 0)\n');
    else
        fprintf('  → This point is not on the root locus (α < 0)\n');
    end
end
```

Q3

```
s = tf('s');
G = 1/((s+2)*(s+3));
K = 2; a = 3; b = 0;
Dc = K*(s+a)/(s+b);
L = minreal(Dc*G);


poles = pole(L);     % [0; -2]
zeros = tzero(L);    % []
n = numel(poles); m = numel(zeros);
nm = n - m;
sigma_a = (sum(poles) - sum(zeros)) / nm;
thetas = ((0:(nm-1))*2 + 1) * pi / nm;
```

```
fig1 = figure('Name','Q3 Root Locus');
rlocus(L); grid on; hold on;
title('Q3 Root Locus with Designed Compensator');


xlim([-6 2]); ylim([-4 4]);


plot(real(zeros), imag(zeros), 'o', 'MarkerSize',8, 'LineWidth',1.5);
plot(real(poles), imag(poles), 'x', 'MarkerSize',10,'LineWidth',1.5);
plot(real(sigma_a), imag(sigma_a), '.', 'MarkerSize',18);


XL = xlim; YL = ylim; R = max([diff(XL), diff(YL)])*1.1;
for th = thetas
    x2 = real(sigma_a) + R*cos(th);
    y2 = imag(sigma_a) + R*sin(th);
    plot([real(sigma_a) x2], [imag(sigma_a) y2], '--', 'LineWidth',1.0);
end


plot([-2 0], [0 0], '-', 'LineWidth',2.0);


plot(-1, 1, 'o', 'MarkerSize',8, 'LineWidth',1.2);
plot(-1,-1, 'o', 'MarkerSize',8, 'LineWidth',1.2);
text(-0.9, 1.1, 'Target -1{+}j', 'FontSize',10);
text(-0.9,-1.1, 'Target -1{-}j', 'FontSize',10);


T = feedback(L, 1);
disp('Closed-loop poles (should include -1±j):');
disp(pole(T));
```