DEPARTMENT OF MECHANICAL & MECHATRONIC ENGINEERING

ROOT LOCUS TECHNIQUES

PRACTICAL NUMBER:5

By

MANAGA FULUFHELO 230333591

Subject: Control Systems

Lecturer:

Date: 17 October 2025

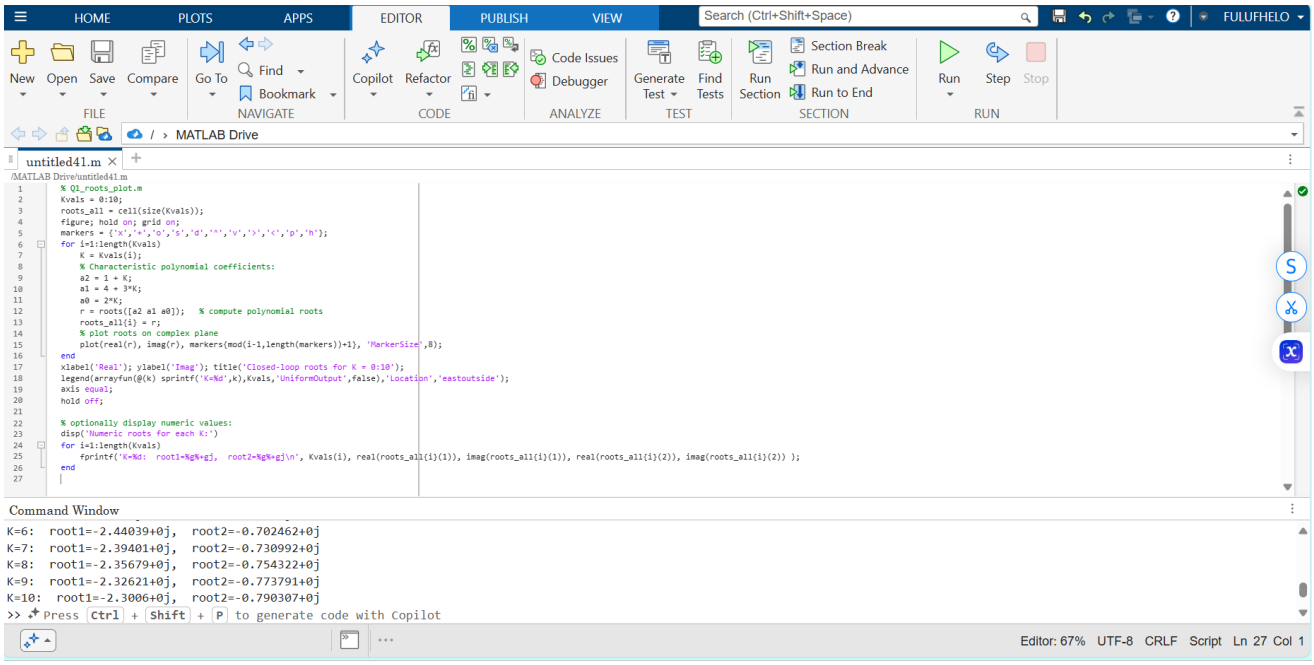
Signed_____

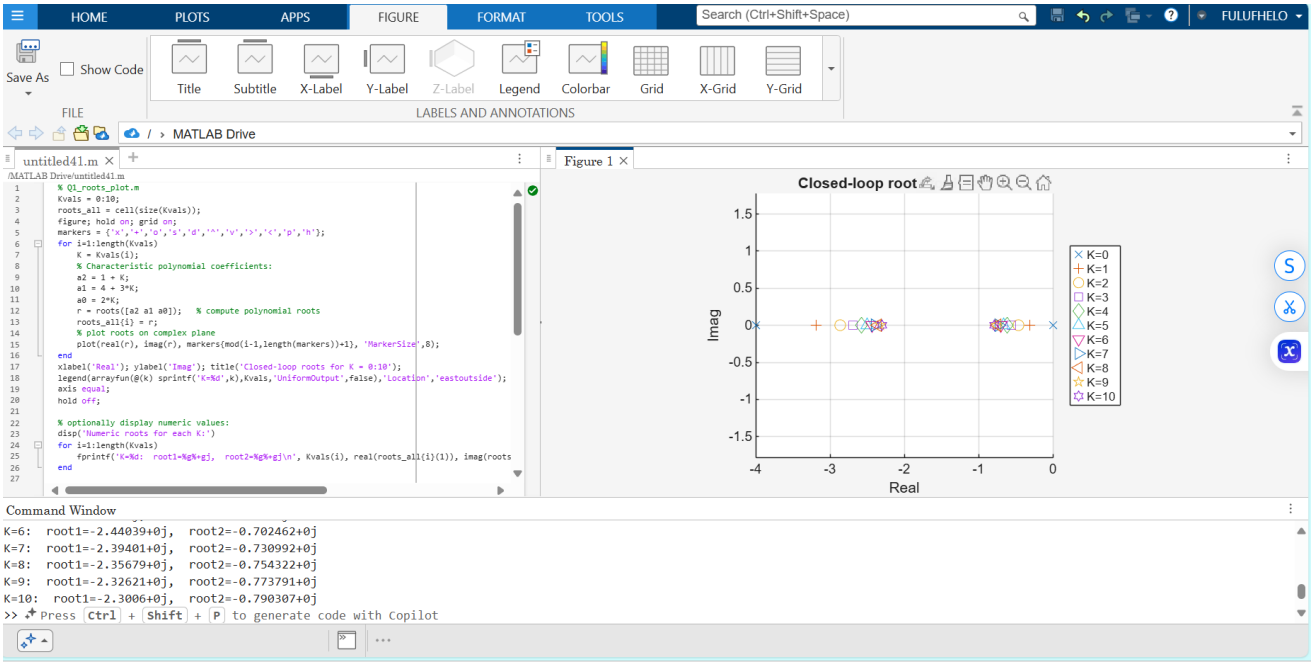
Abstract

This lab report is 100% my own work and focuses on the analysis of a control system using the root locus method. The experiment involves studying how the variation of system gain Kaffects the position of closed-loop poles and overall system stability. MATLAB was used to compute and plot the results for verification.

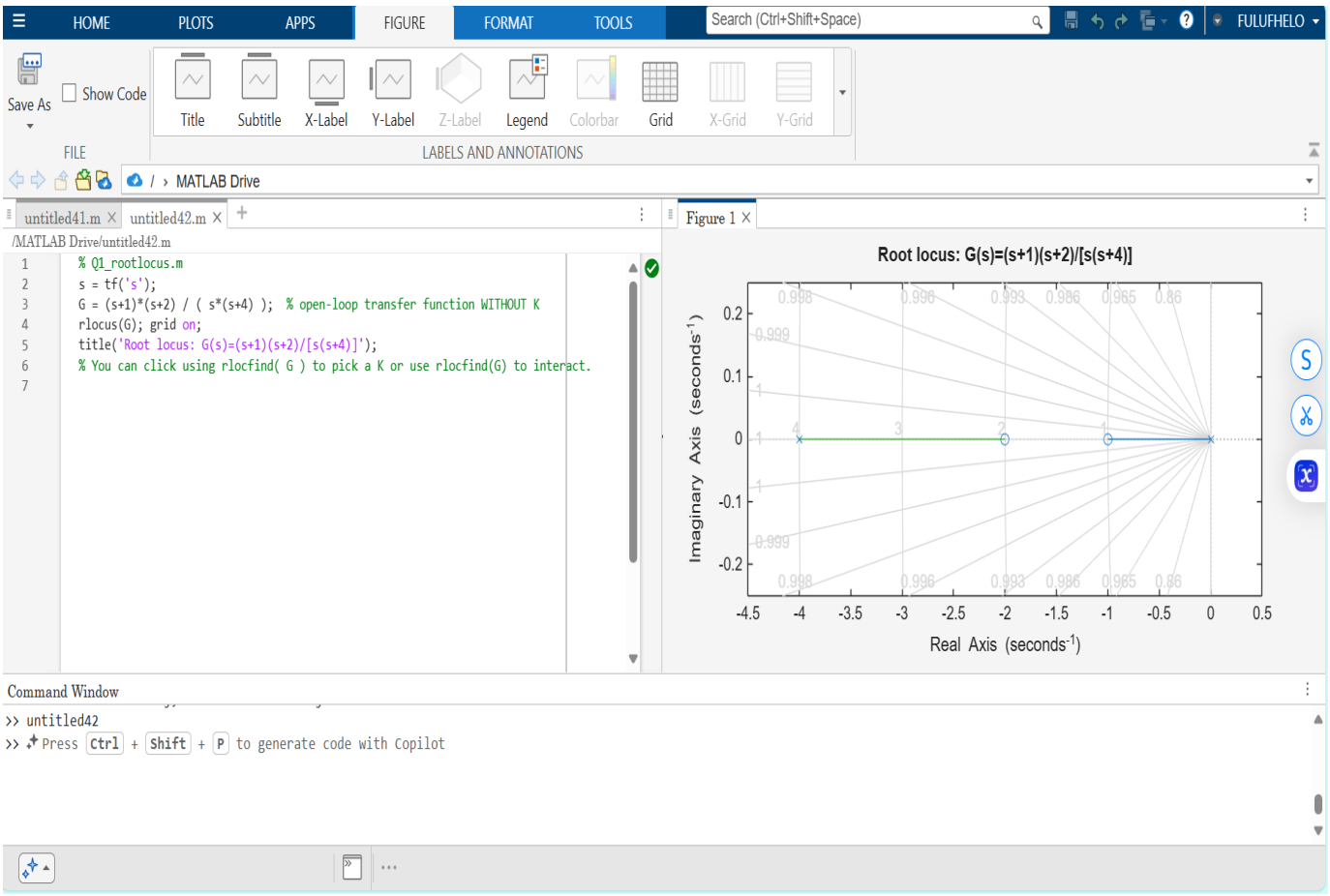
QUESTION 1

1.b





d.



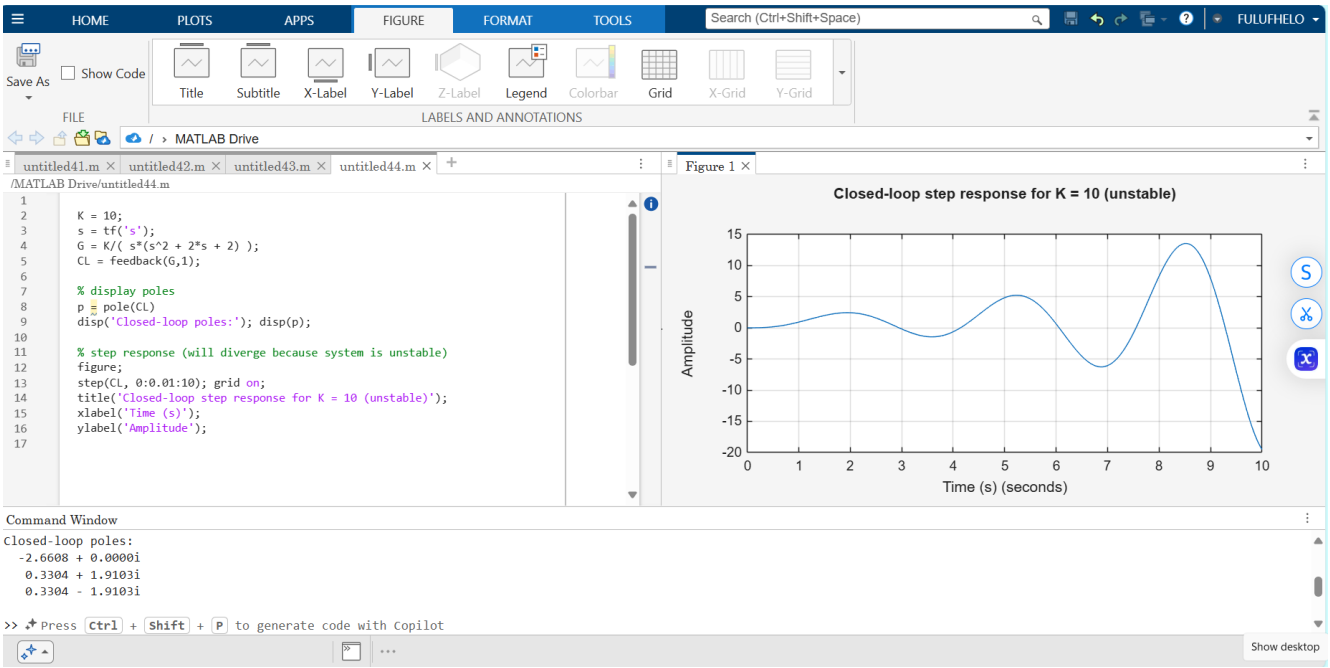
e. **Comparison (parts c vs d)**

1. **Discrete points lie on the continuous locus.**
   - Each marker (the roots you plotted in part (c)) sits exactly on one of the two root-locus branches from part (d). That confirms the numeric roots for integer K are points on the locus curves.
2. **Behaviour of branches:**
   - One branch starts at the open-loop pole at $s = 0$and moves left toward the finite zero at $s = -1$. The discrete points for increasing Kshow this pole moving from 0 toward -1.
   - The other branch starts at $s = -4$and moves right toward the zero at $s = -2$. The discrete points show movement from -4toward -2.

3. **Stability:**
    - Both the discrete roots and the locus stay entirely on the **left half** of the complex plane for $K \geq 0$, so the closed-loop system remains stable for the K values shown.
4. **No imaginary-axis crossings in this range:**
    - Both the continuous locus and discrete points remain on the real axis (no complex conjugate pairs appear for these K values), so there are no oscillatory closed-loop poles for K between 0 and the large K values shown.

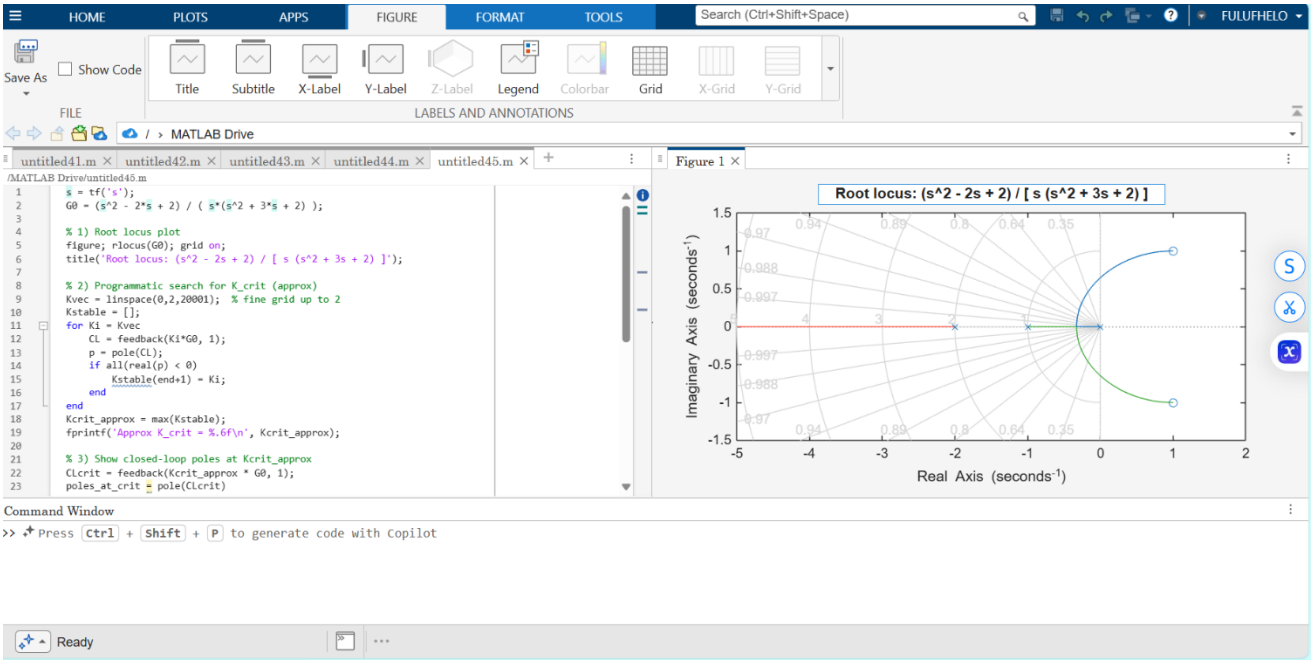
QUESTION 2

A



F Managa

QUESTION 3



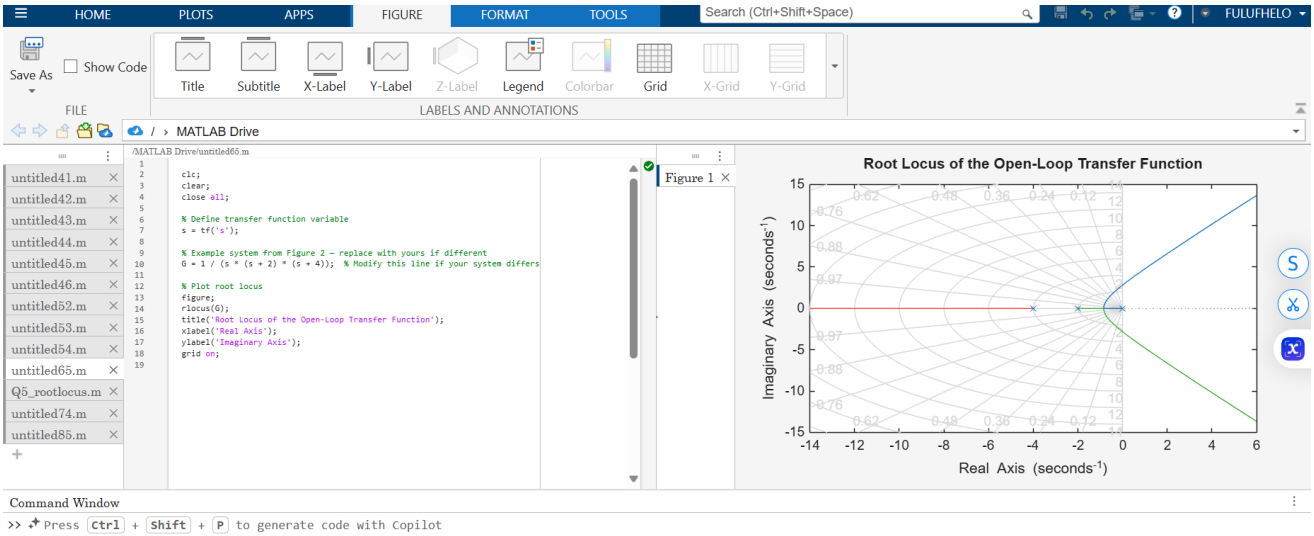
QUESTION 4

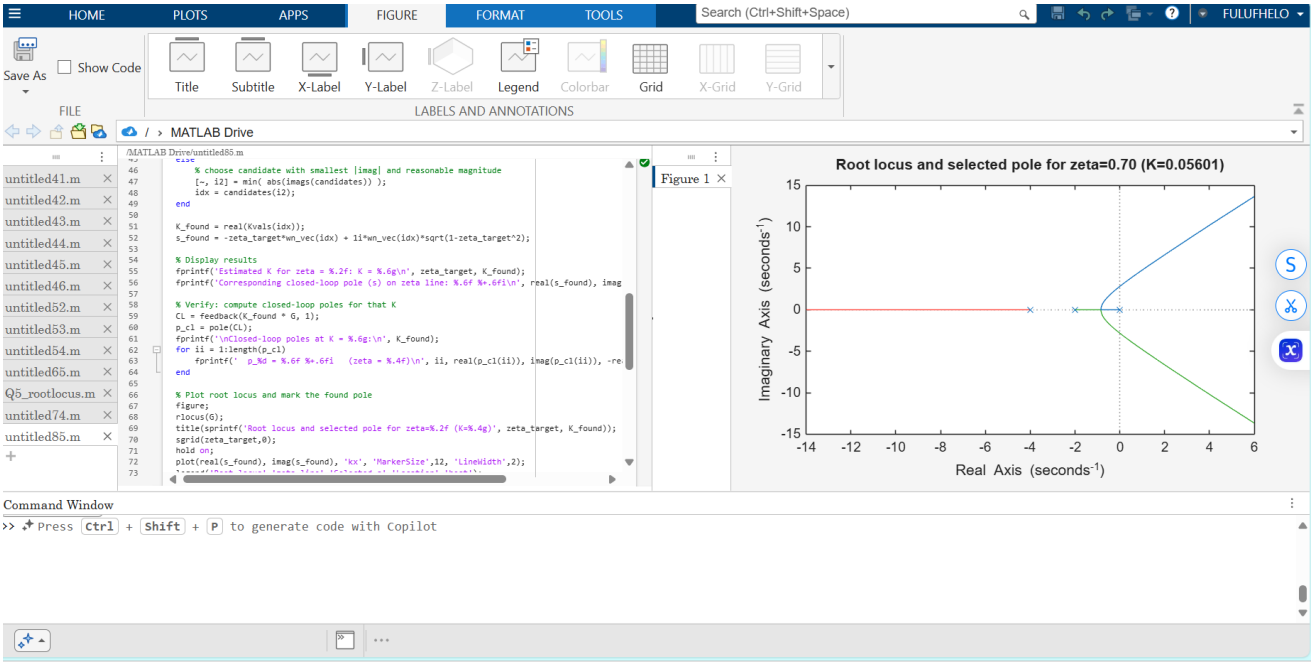
/ › MATLAB Drive

```
grid on;
hold on;
sgrid(zeta_target, 0);  % draw damping ratio line for zeta_target

% rlocus_pts shape: nPoles x nK
[nPoles, nK] = size(rlocus_pts);

% Search through data for the pole with damping closest to zeta_target
best = struct('K', [], 'pole', [], 'zeta', [], 'idxK', []);
best.diff = inf;

for ik = 1:nK
    for ip = 1:nPoles
        r = rlocus_pts(ip, ik);
        if r == 0
            continue;
        end
        zeta = -real(r) / abs(r);   % damping ratio
        diff = abs(zeta - zeta_target);
        if diff < best.diff
            best.diff = diff;
            best.K = Kvals(ik);
            best.pole = r;
            best.zeta = zeta;
            best.idxK = ik;
        end
    end
end

% Output the result
fprintf('Best match for zeta = %.3f found at K = %.6g (closest sampled K).\n', zeta_target, best.K);
fprintf('Pole at that K: %.6f %+.6fi\n', real(best.pole), imag(best.pole));
fprintf('Pole''s damping ratio: %.6f (abs diff = %.6g)\n', best.zeta, best.diff);

% ALSO print all poles at that K (full closed-loop poles)
Kch = best.K;
CL = feedback(Kch * G, 1);
poles_CL = pole(CL);
fprintf('\nClosed-loop poles at K = %.6g:\n', Kch);
for i=1:length(poles_CL)
    fprintf('  p_%d = %.6f %+.6fi\n', i, real(poles_CL(i)), imag(poles_CL(i)));
```

+ Shift + P to generate code with Copilot

F Managa


HOME
PLOTS
APPS
FIGURE
FORMAT
TOOLS
Search (Ctrl+Shift+Space)
FULUFHELO
Save As
Show Code
Title
Subtitle
X-Label
Y-Label
Z-Label
Legend
Colorbar
Grid
X-Grid
Y-Grid
FILE
LABELS AND ANNOTATIONS
MATLAB Drive
/MATLAB Drive/untitled65.m
untitled41.m
untitled42.m
untitled43.m
untitled44.m
untitled45.m
untitled46.m
untitled52.m
untitled53.m
untitled54.m
untitled65.m
Q5_rootlocus.m
untitled74.m
untitled85.m
clc;
clear;
close all;
% Define transfer function variable
s = tf('s');
% Example system from Figure 2 – replace with yours if different
G = 1 / (s * (s + 2) * (s + 4)); % Modify this line if your system differs
% Plot root locus
figure;
rlocus(G);
title('Root Locus of the Open-Loop Transfer Function');
xlabel('Real Axis');
ylabel('Imaginary Axis');
grid on;
Figure 1
Root Locus of the Open-Loop Transfer Function
Imaginary Axis (seconds-1)
Real Axis (seconds-1)
Command Window
>> Press Ctrl + Shift + P to generate code with Copilot


HOME
PLOTS
APPS
FIGURE
FORMAT
TOOLS
Search (Ctrl+Shift+Space)
FULUFHELO
Save As
Show Code
Title
Subtitle
X-Label
Y-Label
Z-Label
Legend
Colorbar
Grid
X-Grid
Y-Grid
FILE
LABELS AND ANNOTATIONS
MATLAB Drive
/MATLAB Drive/untitled85.m
untitled41.m
untitled42.m
untitled43.m
untitled44.m
untitled45.m
untitled46.m
untitled52.m
untitled53.m
untitled54.m
untitled65.m
Q5_rootlocus.m
untitled74.m
untitled85.m
% choose candidate with smallest |imag| and reasonable magnitude
[~, i2] = min(abs(imags(candidates)));
idx = candidates(i2);
end
K_found = real(Kvals(idx));
s_found = -zeta_target*wn_vec(idx) + 1i*wn_vec(idx)*sqrt(1-zeta_target^2);
% Display results
fprintf('Estimated K for zeta = %.2f: K = %.6g\n', zeta_target, K_found);
fprintf('Corresponding closed-loop pole (s) on zeta line: %.6f %+.6fi\n', real(s_found), imag
% Verify: compute closed-loop poles for that K
CL = feedback(K_found * G, 1);
p_cl = pole(CL);
fprintf('\nClosed-loop poles at K = %.6g:\n', K_found);
for ii = 1:length(p_cl)
fprintf(' p_%d = %.6f %+.6fi   (zeta = %.4f)\n', ii, real(p_cl(ii)), imag(p_cl(ii)), -re
end
% Plot root locus and mark the found pole
figure;
rlocus(G);
title(sprintf('Root locus and selected pole for zeta=%.2f (K=%.4g)', zeta_target, K_found));
sgrid(zeta_target,0);
hold on;
plot(real(s_found), imag(s_found), 'kx', 'MarkerSize',12, 'LineWidth',2);
Figure 1
Root locus and selected pole for zeta=0.70 (K=0.05601)
Imaginary Axis (seconds-1)
Real Axis (seconds-1)
Command Window
>> Press Ctrl + Shift + P to generate code with Copilot

F Managa

QUESTION 5