

EENG 5170 Control and Mechatronics

Homework Assignment - 1

Q1 Use Matlab to find the roots of the following polynomials:

Code:

```
Desktop > Matlab assignment Lab 2 EENG5650
Editor - C:\Users\kgaga\Desktop\Matlab assignment Lab 2 EENG5650\EENG_5170_HW_1.m
EENG_5170_HW_1.m x EENG_5170_HW_2.m x EENG_5170_HW_3.m x +
1      clc %clear screen
2      %EENG 5170 HW Assignment 1
3      % Q1. a
4      p1=[2 -3 4 -1 3 0];%polynomial expression
5      r1 = roots(p1) ;%roots
6      disp("Roots for P1 = ")
7      disp(r1)
8
9      %Q1. b
10     p2 =[-1,0,0,5,-1,0,10];%polynomial expression
11     r2 = roots(p2);%roots
12     disp("Roots for P2 = ")
13     disp(r2)
14
15     %Q2
16
```

Result:

```
Editor - C:\Users\kgaga\Desktop\Matlab assignment Lab 2 EENG5650\EENG_5170_
EENG_5170_HW_1.m x EENG_5170_HW_2.m x EENG_5170_HW_3.m x
1      clc %clear screen
2      %EENG 5170 HW Assignment 1
3      % Q1. a
4      p1=[2 -3 4 -1 3 0];%polynomial expression
5      r1 = roots(p1) ;%roots
6      disp("Roots for P1 = ")
7      disp(r1)
8
9      %Q1. b
10     p2 =[-1,0,0,5,-1,0,10];%polynomial expression
11     r2 = roots(p2);%roots
12     disp("Roots for P2 = ")
13     disp(r2)

Command Window

Roots for P1 =
    0.0000 + 0.0000i
    0.9602 + 1.0805i
    0.9602 - 1.0805i
   -0.2102 + 0.8208i
   -0.2102 - 0.8208i

Roots for P2 =
    1.8265 + 0.0000i
   -0.9753 + 1.6199i
   -0.9753 - 1.6199i
    0.6180 + 0.9977i
    0.6180 - 0.9977i
   -1.1119 + 0.0000i
```

Q2 Use Matlab to find the poles and zeros of the following transfer function:

Code:

```
Editor - C:\Users\kgaga\Desktop\Matlab assignment Lab 2 EENG5650\EENG_5170_U_1
EENG_5170_HW_1.m x EENG_5170_HW_2.m x EENG_5170_HW_3.m x
1 % Q2
2 clc
3 num1 = [1 5 -7]; % s^2 + 5s - 7
4 den1 = [1 -1 2 -5 2]; % s^4 - s^3 - 5s + 2
5 sys1 = tf(num1,den1);
6 p = pole(sys1); % pole values
7 z = zero(sys1); % zero values
8 disp("Poles ="); disp(p)
9 disp("Zeros ="); disp(z)
10 pzmap(sys1)
11

Editor - C:\Users\kgaga\Desktop\Matlab assignment Lab 2 EENG5650
EENG_5170_HW_1.m x EENG_5170_HW_2.m x EENG_5170_U_1.m x
1 % Q2
2 clc
3 num1 = [1 5 -7]; % s^2 + 5s - 7
4 den1 = [1 -1 2 -5 2]; % s^4 - s^3 - 5s + 2
5 sys1 = tf(num1,den1);
6 p = pole(sys1); % pole values
7 z = zero(sys1); % zero values
8 disp("Poles ="); disp(p)
9 disp("Zeros ="); disp(z)
10 pzmap(sys1)
11

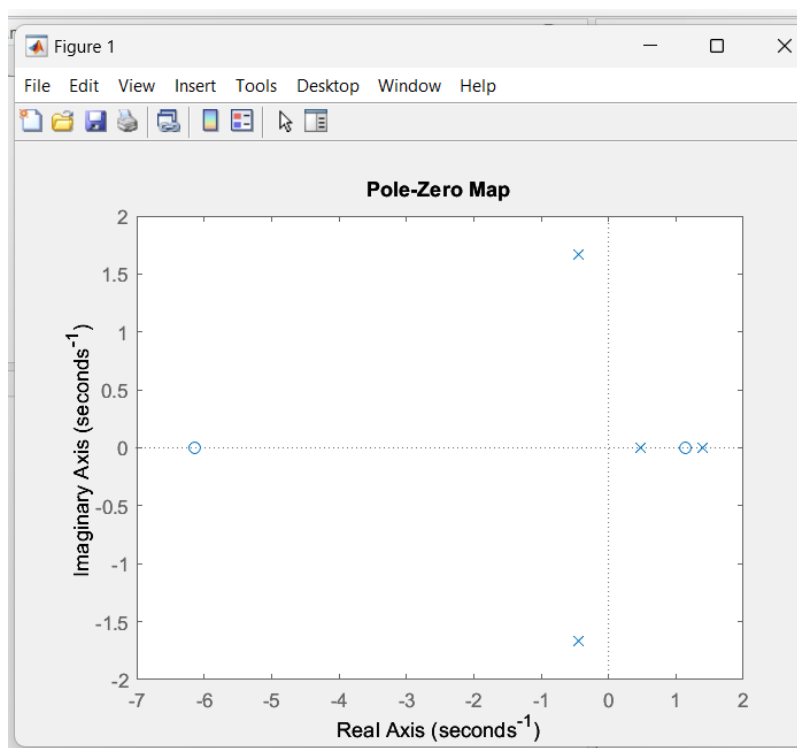
Command Window

Poles =
-0.4389 + 1.6685i
-0.4389 - 1.6685i
1.3969 + 0.0000i
0.4810 + 0.0000i

Zeros =
-6.1401
1.1401
```

Result:

Plot:



Q3

Code:

Desktop ▶ Matlab assignment Lab 2 EENG5650

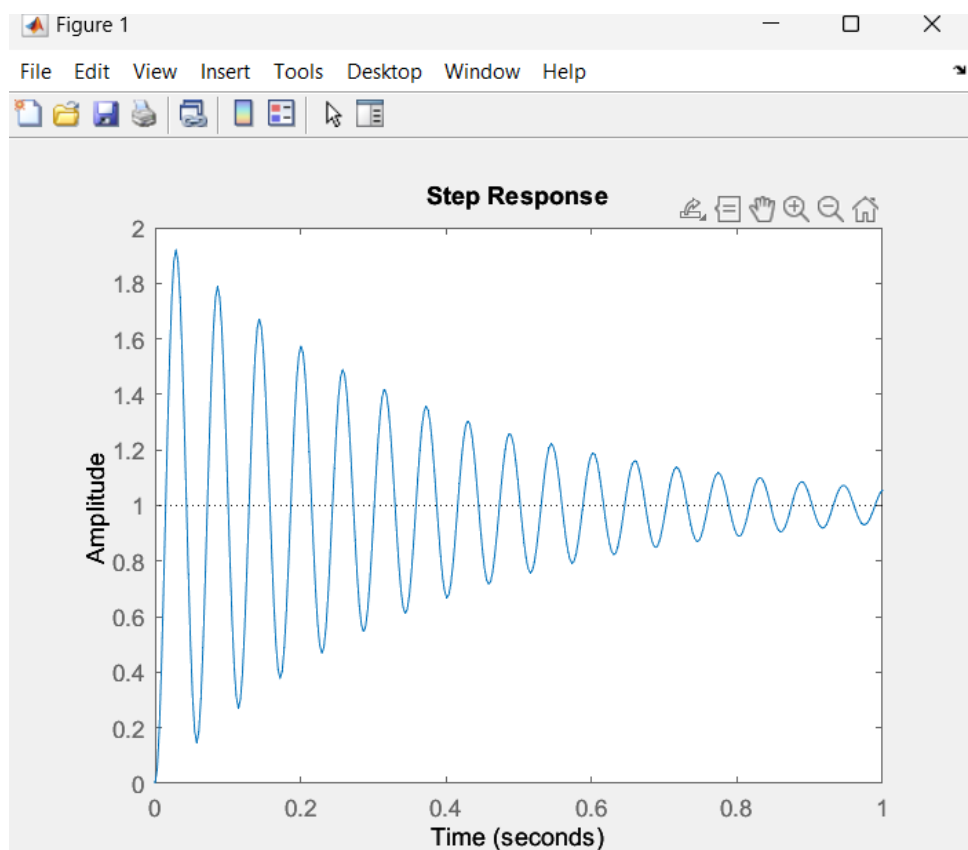
Editor - C:\Users\kgaga\Desktop\Matlab assignment Lab 2 EENG5650\EENG_5170_HW_3.m

```
EENG_5170_HW_1.m x EENG_5170_HW_2.m x EENG_5170_HW_3.m x +
1 %EENG 5170 Q3
2 clc
3 g1 = tf([20],[1,2]);
4 g2 = tf([1],[1,3.5]);
5 sys1 = g1*g2;
6 sys1_1 = feedback(sys1, 0.2, -1);
7 g3 = 600;
8 sysR = g3*sys1_1;
9 sys = feedback(sysR, 1, -1);
10 step(sys, 1);
```

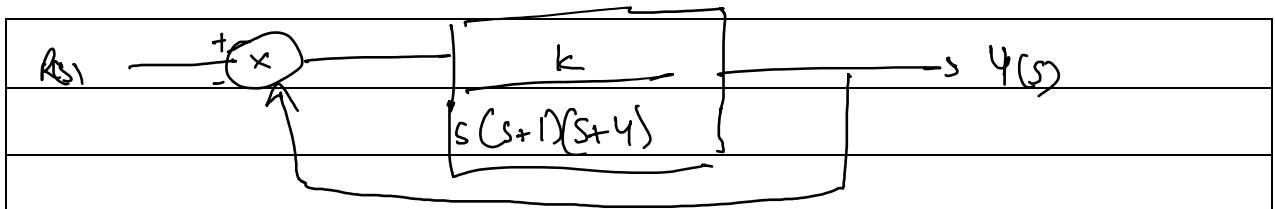
Name ^	Value
den1	[1,-1,2,-5,2]
g1	1x1 tf
g2	1x1 tf
g3	600
num1	[1,5,-7]
out	1x1 SimulationOutput
p	[-0.4389 + 1.6685i;-0.4389 - 1.668...
p1	[2,-3,4,-1,3,0]
p2	[-1,0,0,5,-1,0,10]
r1	[0.0000 + 0.0000i;0.9602 + 1.0805i...
r2	[1.8265 + 0.0000i;-0.9753 + 1.619...
sys	1x1 tf
sys1	1x1 tf
sys1_1	1x1 tf
sysR	1x1 tf
z	[-6.1401;1.1401]

Workspace:

Plot:



Q4:



a] Closed loop transfer function.

$$E(s) = R(s) - Y(s) \quad Y(s) = G(s) - E(s) \quad Y(s) = G(s) - E(s)$$

$$\therefore \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad H(s) = 1$$

$$\therefore \frac{Y(s)}{R(s)} = \frac{k}{s(s+1)(s+4)} = \frac{k}{(s)(s+1)(s+4) + k}$$

$$1 + k \frac{s(s+1)(s+4)}{s(s+1)(s+4)} \therefore \frac{Y(s)}{R(s)} = \frac{k}{s^3 + 5s^2 + 4s + k}$$

b] Routh - Hurwitz Stability Criterion

$$a_n \neq 0 \quad a_0 \neq 0 \quad D(s) = s^3 + 5s^2 + 4s + k$$

\therefore Routh Table

s^3	1	4	$C_1 = -\left[\frac{k-20}{5}\right]$
s^2	5	k	
	$\frac{20-k}{5}$	0	
	k		

$C_2 = -\left[\frac{0-0}{5}\right]$

$d_1 = \frac{1}{5} \left[0 + k \left(\frac{20-k}{5} \right) \right]$

$\left(\frac{20-k}{5} \right)$

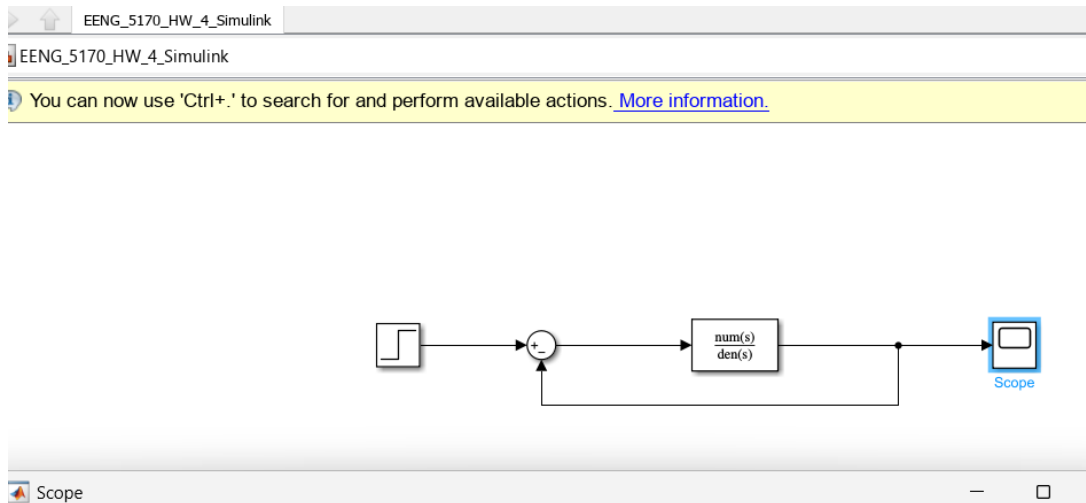
No sign change in 1st column

$$\therefore \frac{20-k}{5} \geq 0 \quad \therefore k > 0$$

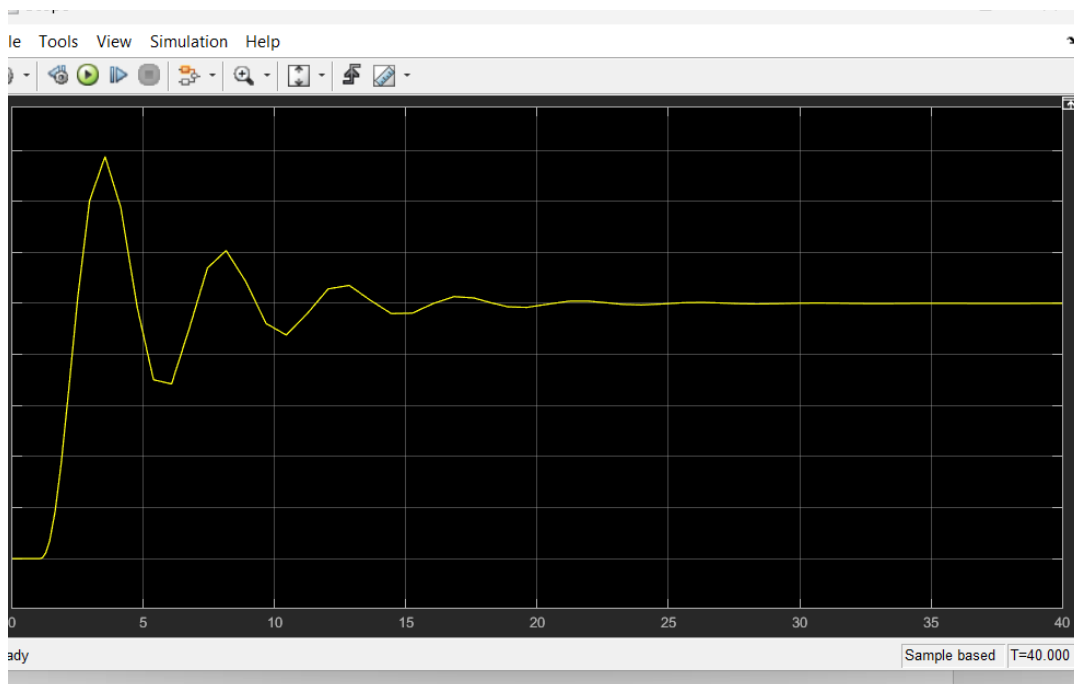
\therefore System is stable if

$$0 < k < 20$$

Simulink:



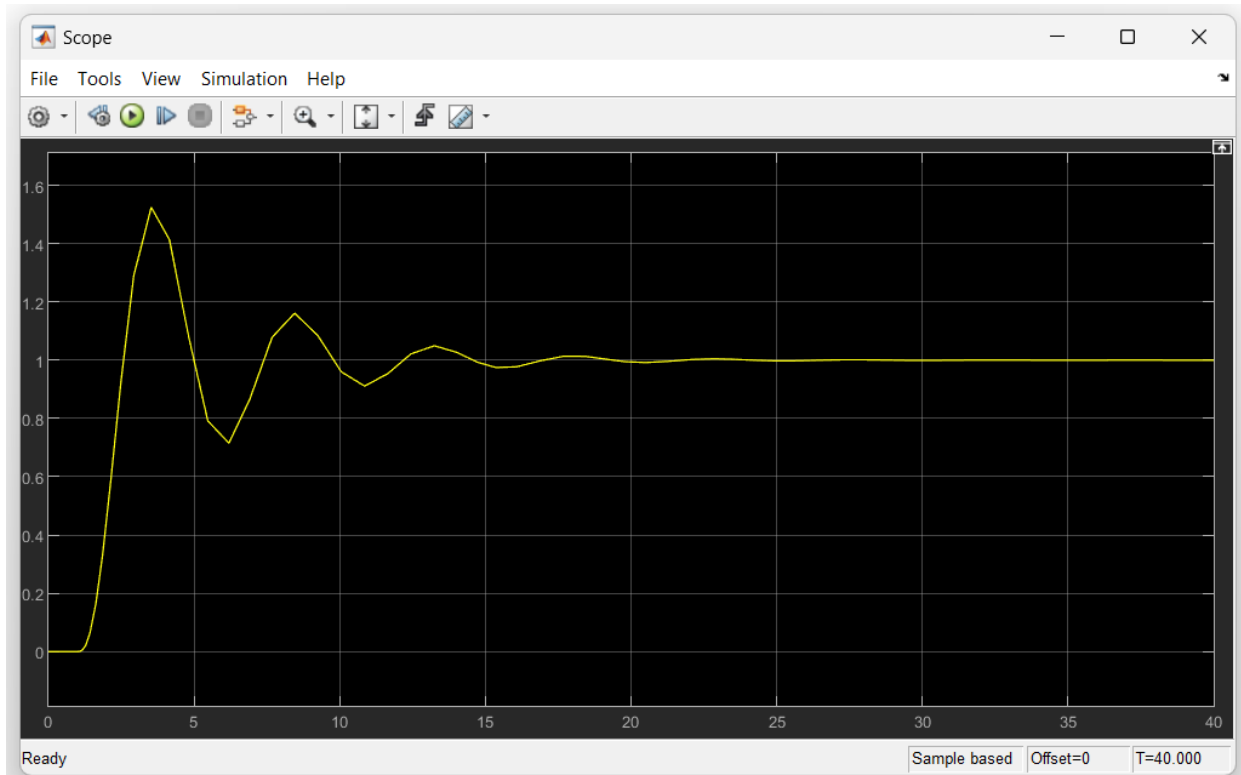
For $k = 3$: $t = 40$



Stable!

For $k = 8$ $t = 40$

Stable!



$k = -2$ $t = 40$

Block Parameters: Transfer Fcn

Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s .

'Parameter tunability' controls the runtime tunability level for numerator and denominator coefficients.
 'Auto': Allow Simulink to choose the most appropriate tunability level.
 'Optimized': Tunability is optimized for performance.
 'Unconstrained': Tunability is unconstrained across the simulation targets.

Parameters

Numerator coefficients: [-2] -2

Denominator coefficients: [1 5 4 0] [1,5,4,0]

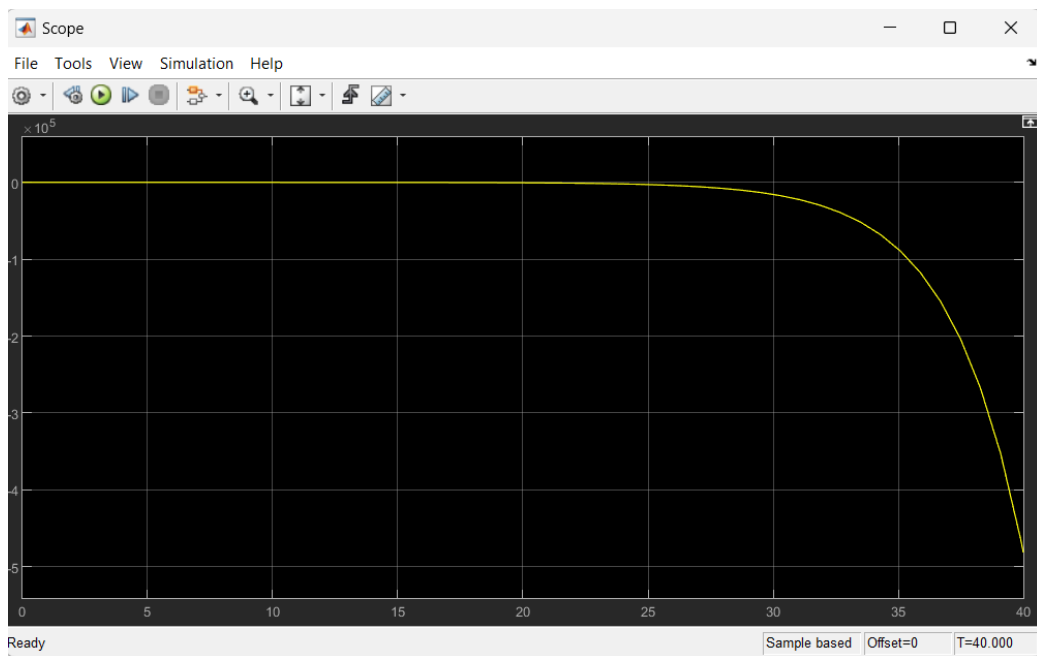
Parameter tunability: Auto

Absolute tolerance: auto

State Name: (e.g., 'position') Absolute tolerance: (Name: AbsoluteTolerance)

OK Cancel Help Apply

Not stable



$K = 23$ $t = 40$

Not stable

