Elec341 - Systems and Control - Project assignment - Abhishek Raghuwanshi (13720222)

Note: all plots are included in this pdf and can also be generated using the matlab and simulink file submitted with this.

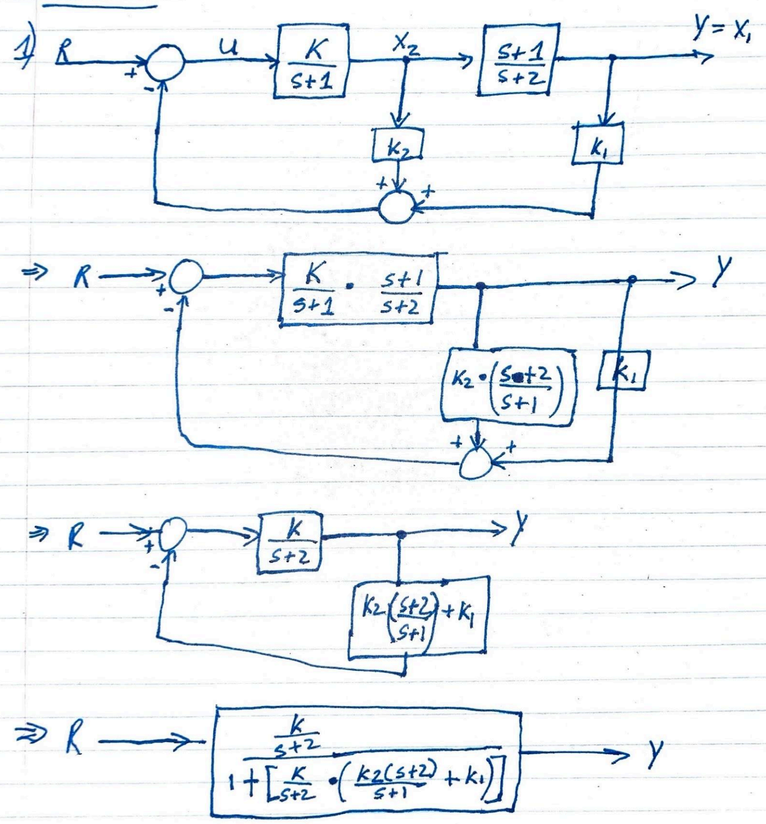
Part I

1)

Abhishek Raghuwanshi
13720222

Elec 341 Project

Part-I



$\therefore$ Transfer function is: $Y(s)/R(s) = T(s) = \dfrac{\frac{K}{s+2}}{1 + \frac{K k_2 (s+2) + K k_1 (s+1)}{(s+1)(s+2)}}$

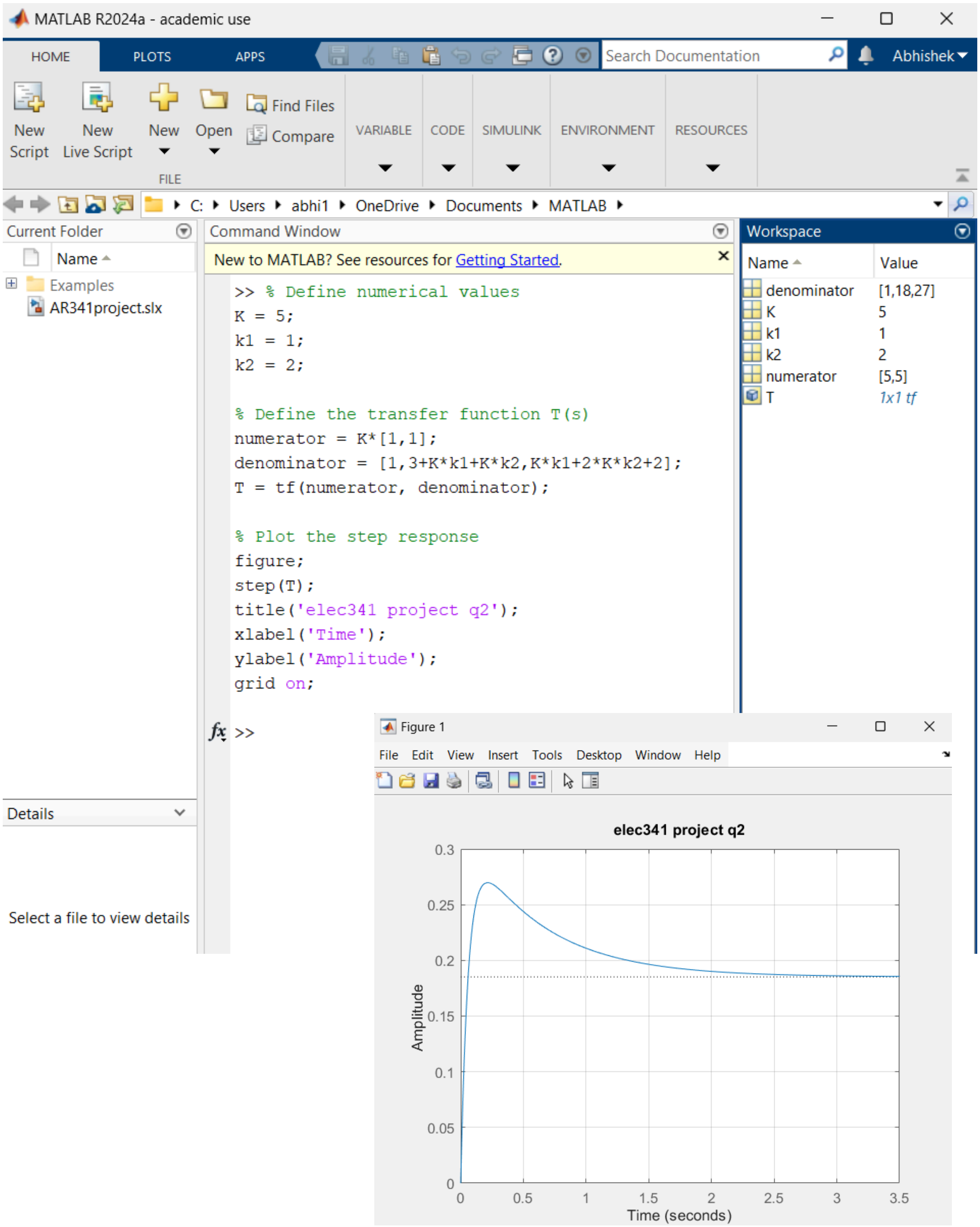
$\Rightarrow \dfrac{K/(s+2)}{\frac{(s+1)(s+2) + K k_2 (s+2) + K k_1 (s+1)}{(s+1)(s+2)}}$

$\Rightarrow \boxed{T(s) = \dfrac{K(s+1)}{(s+1)(s+2) + K k_2 (s+2) + K k_1 (s+1)}}$

Simplifying further I got T(s) = [K(s+1)] / [s^2 + (K*k1+K*k2+3)*s + (K*k1+2*K*k2+2)].

Elec341 - Systems and Control - Project assignment - Abhishek Raghuwanshi (13720222)

2) Note: I have also copypasted the matlab code for all questions as reference on the last page of this pdf for convenience.



3)

3) Need to find condition for closed loop system to follow step input with no steady-state error.

$$\mathcal{E}_{ss} = \lim_{t\to\infty} (r(t) - y(t)).$$

(steadystate error)

since ~~r(t) is 1st~~ r(t) is ~~a~~ step unit input,

$$\lim_{t\to\infty} r(t) = 1.$$

∴ to find $\mathcal{E}_{ss} = 0$,

$$0 = \mathcal{E}_{ss} = \lim_{t\to\infty} (1 - y(t)) = \lim_{s\to 0} \left(\frac{1}{s} - Y(s)\cdot R(s)\right)$$

$$\Rightarrow 0 = \lim_{s\to 0} \left(\frac{1}{s} - Y(s)\cdot R(s)\right) \quad \text{and } Y(s) = T(s)\cdot R(s)$$

$$\Rightarrow \lim_{s\to 0} \frac{K(s+1)\cdot \cancel{1/s}}{s^2 + (KK_1 + KK_2 + 3)s + KK_1 + 2KK_2 + 2} = \cancel{1/s}$$

[it can cancel out since $s \neq 0$, only $s \to 0$]

$$\Rightarrow \frac{K}{KK_1 + 2KK_2 + 2} = 1$$

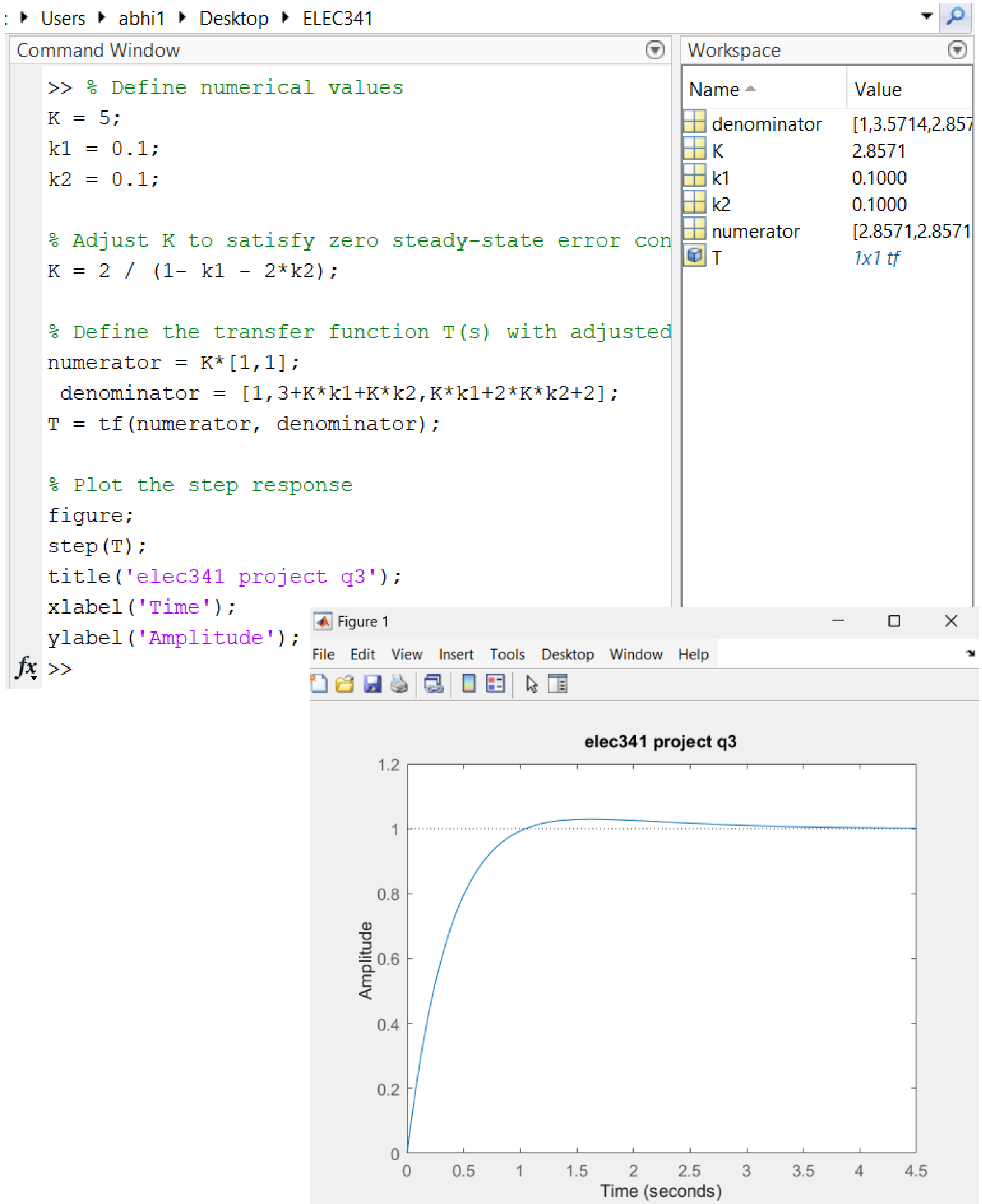
$$\Rightarrow K - KK_1 - 2KK_2 = 2$$

$$K = \frac{2}{1 - K_1 - 2K_2}$$

For zero steady state error, limit of T(s) at s->0 = 1,

So, choose values of K, k1 and k2 that satisfy this condition, in this case by algebraically modifying the T(s) at s->0 = 1 to separate out K, we find K = 2 /(1-k1-2*k2)

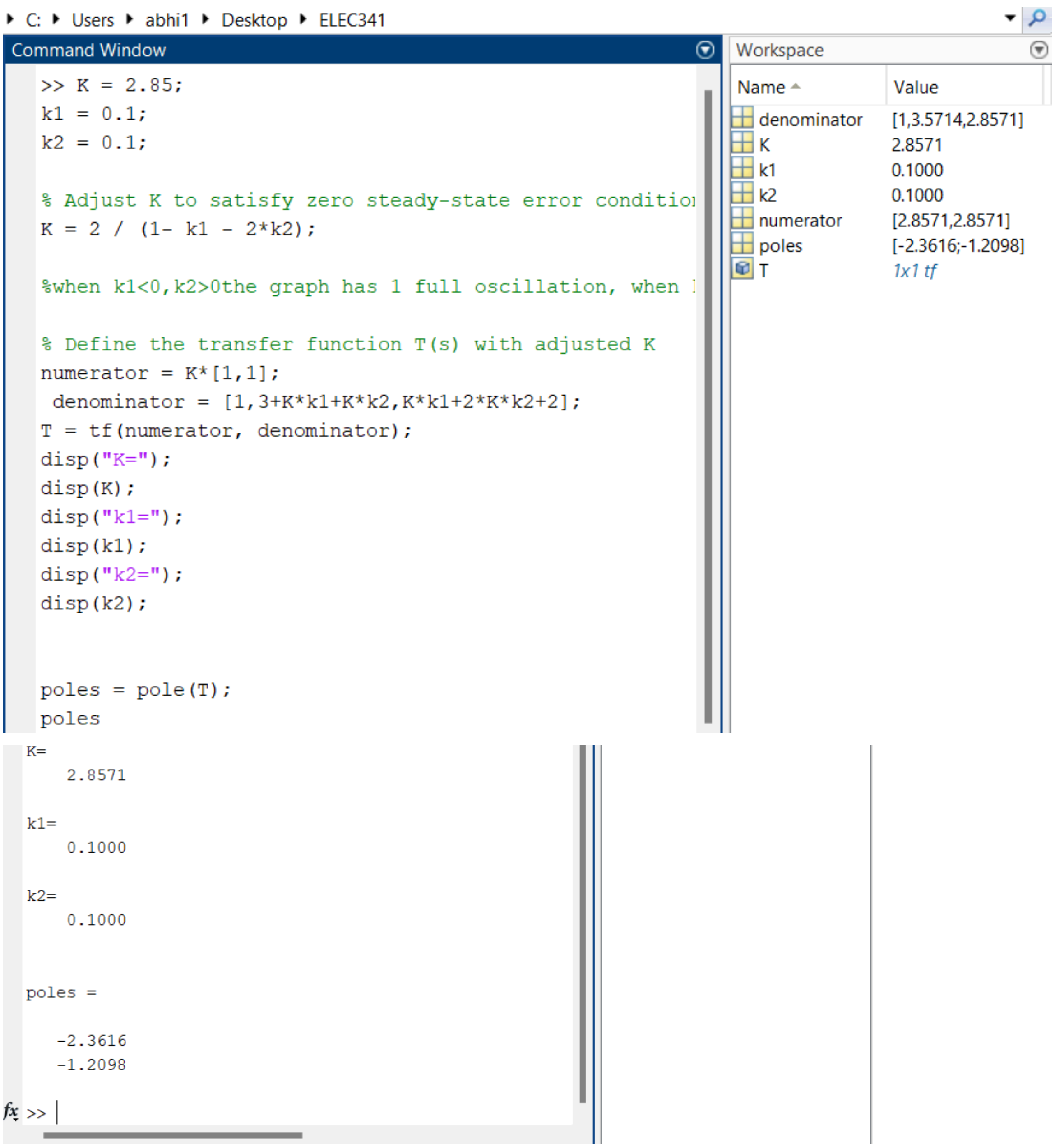
That gives K = 2.85,when k1=0.1 and k2=0.1 as shown in the workspace.
Playing around with values of k1 and k2, when k1<0,k2>0the graph has 1 full oscillation, when k1=0.1,k2=0.1, half oscillation but always evens out at 1 (steady state error=0).



4) The poles of the closed-loop transfer function T(s) are given by the roots of the denominator polynomial.

The denominator polynomial is:
s^2 + (K*k1+K*k2+3)*s + (K*k1+2*K*k2+2)

Therefore to solve for poles, denominator=0 and solve for s:
Since the denominator can’t be fully factored without having values of the parameters,
I used the 0 steady state error values from q3.



This gave me values for poles using the pole() function on matlab.

Therefore at k1=0.1 and k2=0.1, K=2.8571 and poles are:
-2.3616, -1.2098

We can also observe that parameters k1 and k2 indeed can be used to control position of both poles since K directly multiplies with (s+1) and K is a function of k1 and k2 to ensure steady error=0, thereby making it so that both k1 and k2 affect the position of poles.

On actually changing these parameter values we see that:
-In the TF, k1 and k2 occur in the denominator, meaning that pole locations can be controlled by changing k1 and k2.
-The denominator can't be fully factored, but it can be seen that the pole at s = -2 doesn't change with k1 and k2

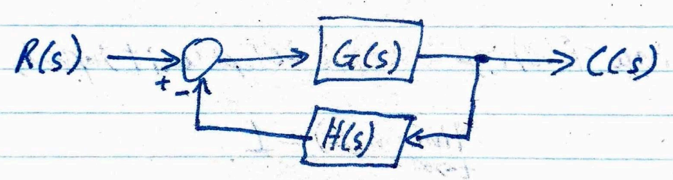
So yes, k1 and k2 can be used to control the positions of the poles.

PART - II

5) need to compute overall transfer function analytically:

Part 2:

5) $G(s) = \dfrac{K_1}{s\left(\frac{s^2}{2600}+\frac{s}{26}+1\right)}$, $H(s) = \dfrac{1}{0.04s+1}$



$$\therefore T(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1+(G(s)\cdot H(s))}$$

$$\Rightarrow T(s) = \frac{2600\cdot K_1\cdot(0.04s+1)}{(s^3+100s^2+2600s)(0.04s+1)+2600K_1}$$

6)

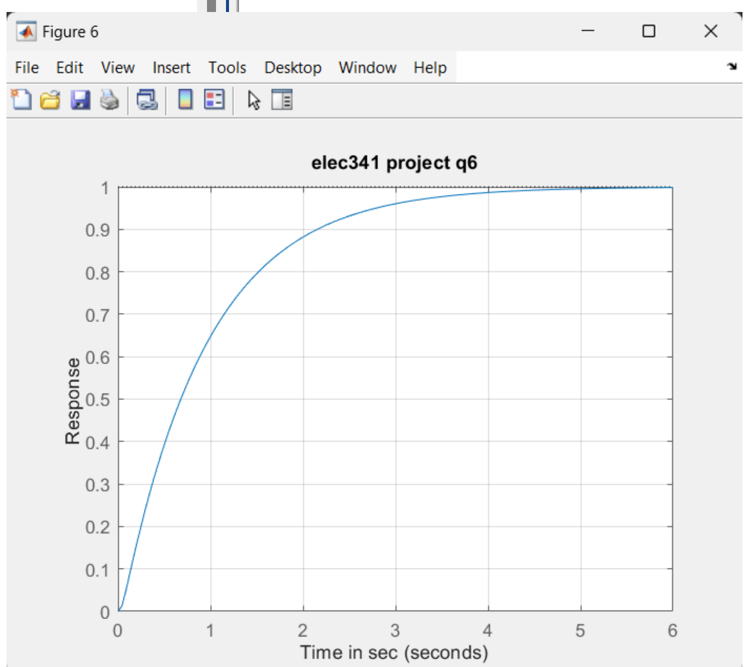
C: ▸ Users ▸ abhi1 ▸ Desktop ▸ ELEC341

Command Window

```
>> K1 = 1;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);

%closed loop transfer function:G*H
GH= series(G,H);
%open loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final open loop transfer function is");
T

%simulate and plot unit step response:
figure;
step(T);
title('elec341 project q6');
xlabel('Time in sec');
ylabel('Response');
grid on;
```

Workspace

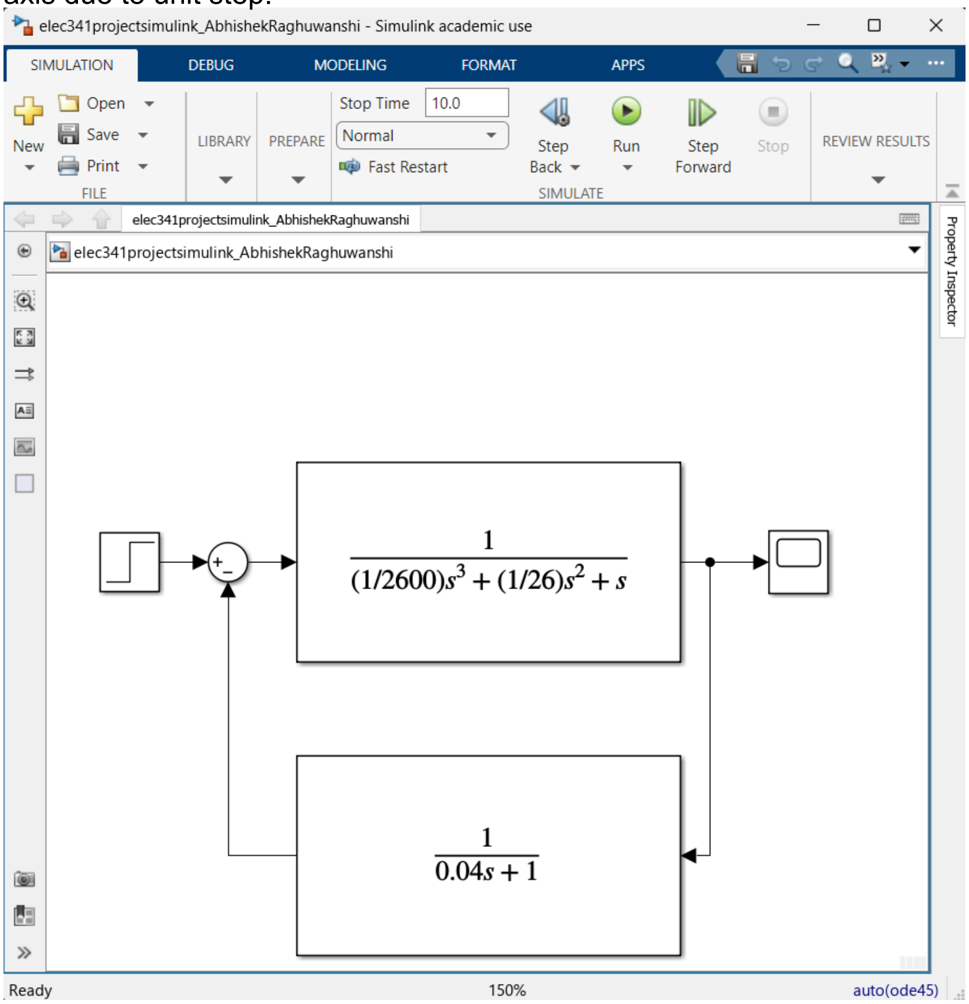
Name	Value
G	1x1 tf
G_den	[3.8462e-04,0.03...
G_num	1
GH	1x1 tf
H	1x1 tf
H_den	[0.0400,1]
H_num	1
K1	1
T	1x1 tf

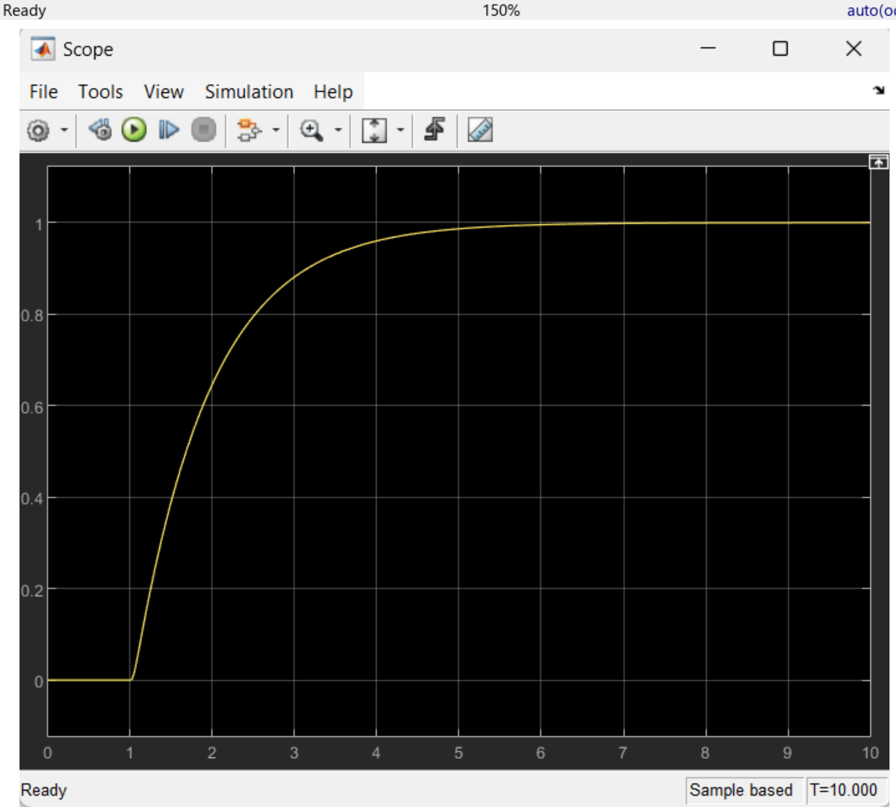


Resulting tf is printed as below
and plot unit response is shown on right.

```
G_den =

    0.0004    0.0385    1.0000         0

Closed loop transfer function is

T =

                    0.04 s + 1
  ------------------------------------------------
  1.538e-05 s^4 + 0.001923 s^3 + 0.07846 s^2 + s + 1

Continuous-time transfer function.
Model Properties
>>
```

7) We can see that the simulink scope plot matches the matlab plot with an offset of 1 along x axis due to unit step.





I personally find simulink to be easier to use since it is simply inserting blocks to the diagram and checking scope by running simulation, however I enjoy matlab more since I feel I get a better understanding of the system while typing the commands and functions.

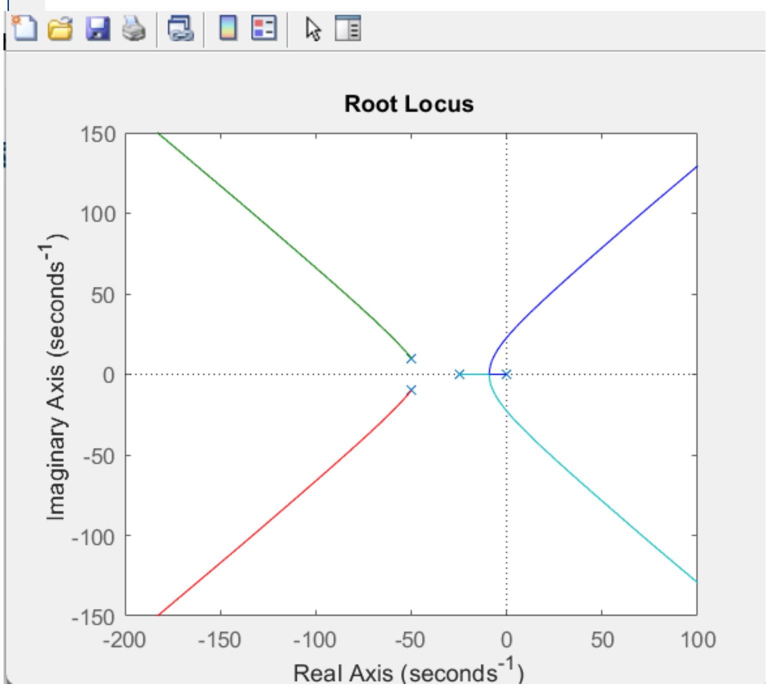
8) Use matlab to plot rootlocus of closed loop system (GH in this case).

Command Window

```
>> K1 = 1;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);

%closed loop transfer function:G*H
GH= series(G,H);
%open loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final open loop transfer function is");
T
%need root locus plot of close loop system(GH)
rlocus(GH);
```

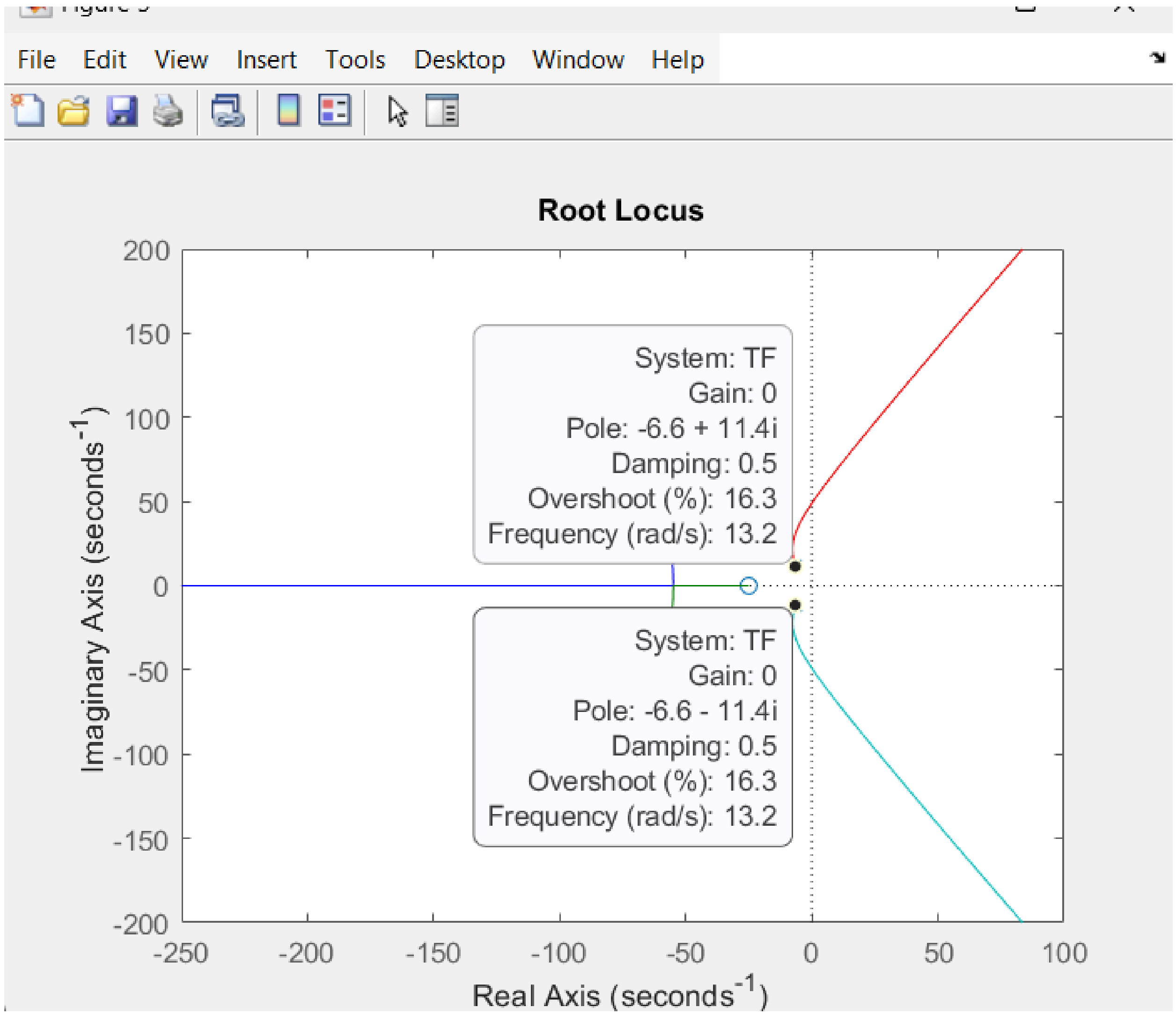
Workspace

Name	Value
G	1x1 tf
G_den	[3.8462e-04,0.03...
G_num	1
GH	1x1 tf
H	1x1 tf
H_den	[0.0400,1]
H_num	1
K1	1
T	1x1 tf



We find that, for K1 > 0, all poles occur on the LHP of the root locus plot (negative real part). This means that the TF for K1 > 0 is stable.

9) C(s)/R(s) is the final open loop transfer function T(s).
To find value of K1 with damping factor zeta=0.5, plot root locus of T and use slider to damping = 0.5. This works at value of K1 = 9.25.
Values at dominant root:
Damping Zeta = 0.5 at roots = -6.60 + 11.4j and -6.60 - 11.4j for K1 = 9.25.



Other remaining roots occur at -55.9 +- 18j with damping Zeta = 0.952

10) Define value of K1 as found in previous question and evaluate transfer function.

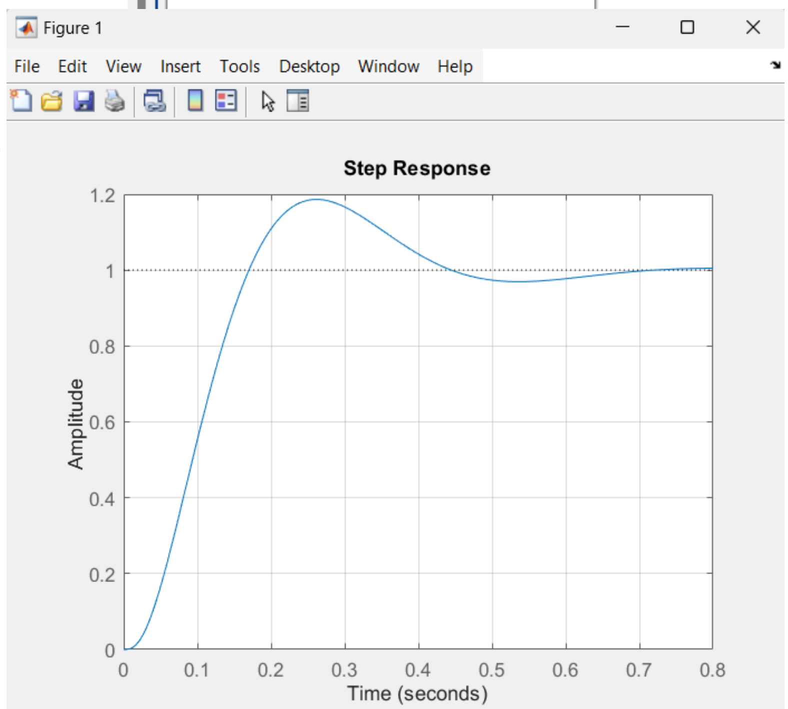
› C: › Users › abhi1 › Desktop › ELEC341

Command Window

```
>> K1 = 9.25;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);

%open loop transfer function:G*H
GH= series(G,H);
%closed loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final transfer function at K=9.25 is");
T
%unit step response:
step(T);
grid on;
```

Workspace

Name	Value
G	1x1 tf
G_den	[3.8462e-04,0.03...
G_num	9.2500
GH	1x1 tf
H	1x1 tf
H_den	[0.0400,1]
H_num	1
K1	9.2500
T	1x1 tf



The step response is as shown on right
With damping factor 0.5

The transfer function at K1=9.25 is as below

```
G_den =

    0.0004    0.0385    1.0000         0

Final transfer function at K=9.25 is

T =

                      0.37 s + 9.25
  -----------------------------------------------------
  1.538e-05 s^4 + 0.001923 s^3 + 0.07846 s^2 + s + 9.25

Continuous-time transfer function.
Model Properties
fx >>
```

<u>ALL MATLAB CODE:</u>

<u>Part 1:</u>

<u>Q2 matlab code</u>

```
>>% Define numerical values
K = 5;
k1 = 0.1;
k2 = 0.1;

% Define the transfer function T(s)
numerator = K*[1,1];
denominator = [1,3+K*k1+K*k2,K*k1+2*K*k2+2];
T = tf(numerator, denominator);

% Plot the step response
figure;
step(T);
title('elec341 project q2');
xlabel('Time');
ylabel('Amplitude');
grid on;
```

<u>Q3 matlab code</u>

```
>> % Define numerical values
K = 5;
k1 = 0.1;
k2 = 0.1;

% Adjust K to satisfy zero steady-state error condition
K = 2 / (1- k1 - 2*k2);

%when k1<0,k2>0the graph has 1 full oscillation, when k1=0.1,k2=0.1, half oscillation that evens out at 1.

% Define the transfer function T(s) with adjusted K
numerator = K*[1,1];
 denominator = [1,3+K*k1+K*k2,K*k1+2*K*k2+2];
T = tf(numerator, denominator);

% Plot the step response
```

```
figure;
step(T);
title('elec341 project q3');
xlabel('Time');
ylabel('Amplitude');
grid on;
```

<u>Q4 matlab code</u>

```
>> % Define numerical values
K = 2.85;
k1 = 0.1;
k2 = 0.1;

% Adjust K to satisfy zero steady-state error condition
K = 2 / (1- k1 - 2*k2);

%when k1<0,k2>0the graph has 1 full oscillation, when k1=0.1,k2=0.1, half oscillation that
evens out at 1.

% Define the transfer function T(s) with adjusted K
numerator = K*[1,1];
 denominator = [1,3+K*k1+K*k2,K*k1+2*K*k2+2];
T = tf(numerator, denominator);
disp(‘K=’);
disp(K);
disp(‘k1=’);
disp(k1);
disp(‘k2=’);
disp(k2);

poles = pole(T);
poles
% Values will be printed below.
```

Part 2:

Q6 matlab code:

```
 K1 = 1;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);

%closed loop transfer function:G*H
GH= series(G,H);
%open loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final open loop transfer function is");
T

%simulate and plot unit step response:
figure;
step(T);
title('elec341 project q6');
xlabel('Time in sec');
ylabel('Response');
grid on;
```

Q7: simulink

Q8 matlab code:

```
K1 = 1;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);
```

```
%closed loop transfer function:G*H
GH= series(G,H);
%open loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final open loop transfer function is");
T
%need root locus plot of close loop system(GH)
rlocus(GH);
```

Q9 matlab code:

```
K1 = 9.25;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);

%open loop transfer function:G*H
GH= series(G,H);
%closed loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final open loop transfer function is");
T
%need root locus plot of close loop system(GH)
rlocus(T);
```

Q10 matlab code:

```
K1 = 9.25;
G_num=K1;
G_den=[1/2600,1/26,1,0]
G=tf(G_num,G_den);

H_num=1;
H_den=[0.04,1];
H=tf(H_num,H_den);
```

```
%open loop transfer function:G*H
GH= series(G,H);
%closed loop transfer function: G/(1+G*H)
T=feedback(G,H);

%Print the transfer function with K=1
disp("Final transfer function at K=9.25 is");
T
%unit step response:
step(T);
grid on;
```

The end :)