Roll: 1715006 Name: Adowan Shahriar LAB: 5

Objectives:

- To be able to find design a PID controller for system.
- To find transient parameters of PID controller
- To be able to calculate phase of PD compensated system and all by MATLAB.
- To be able to calculate steady-state error parameters.

Introduction :-

PID controller is a proportional-integral-derivative controller which controls loop mechanism employing feedback that is widely used in industrial control systems and variety of other applications requiring continuously modulated control.

Root locus, a graphical presentation of the closed loop poles as a system parameter is varied, is a powerful method of analysis and design for stability and transient response. Feedback control systems are difficult to compare comprehend from a qualitative point of view, and hence they rely heavily upon the mathematics. The root locus is a graphical technique that gives us the qualitative tools that yields more information than the methods already discussed. The root locus's real power lies in its ability to provide solutions for systems of order higher than but it also can yield a desired transient response for first order and second order systems.

The root locus can be used to describe qualitatively the performance of a system as various parameters are changed.

Besides transient response the root locus also gives a graphical representation of a system's stability. We can easily see ranges of stability, ranges of instability and the conditions that cause a system to break into oscillation.

A PID controller is shown in figure below. Its transfer function is

$$G_c(s) = k_1 + \frac{K_2}{s} + K_3 s = \frac{K_1 s + K_2 + K_3 s^2}{s}$$

$$= \frac{K_3\left(s^2 + \frac{K_1}{K_3} s + \frac{K_2}{K_3}\right)}{s}$$

Which has two zeros plus a pole at the origin. One zero and a pole at the origin can be designed as the ideal integral compensator; the other zero can be designed as the ideal derivative compensator.

The design technique:

1. Evaluating the performance of the uncompensated system to determine how much improvement in transient response is required.
2. Design the PD controller to meet the transient response specifications. The design includes the zero location and loop gain.

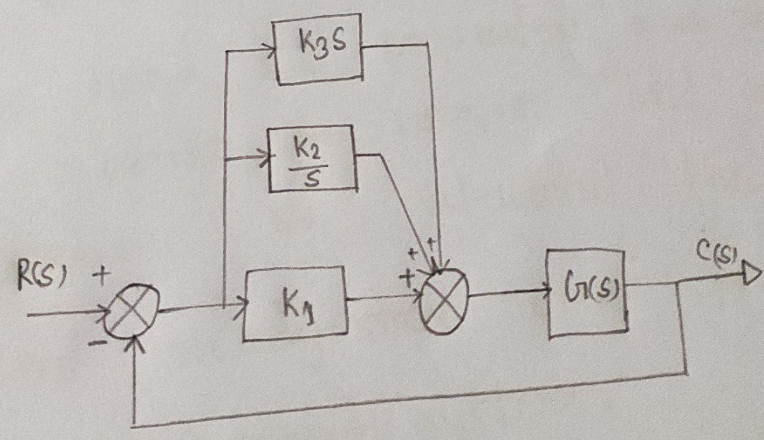


Figure: PID controller

Task:

Editor - C:\Users\DELL\Untitled.m

Untitled.m

```
clear all
close all
clc
s= -8.157+15.92i;
Gs=1*(s+8)/((s+3)*(s+6)*(s+10));
phase=angle(Gs);
Theta=180/pi*phase
```

Output:

Command Window

```
Theta =

  161.5037
```

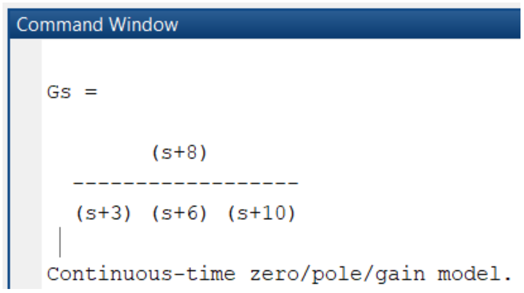
Task:

Editor - C:\Users\DELL\Untitled.m

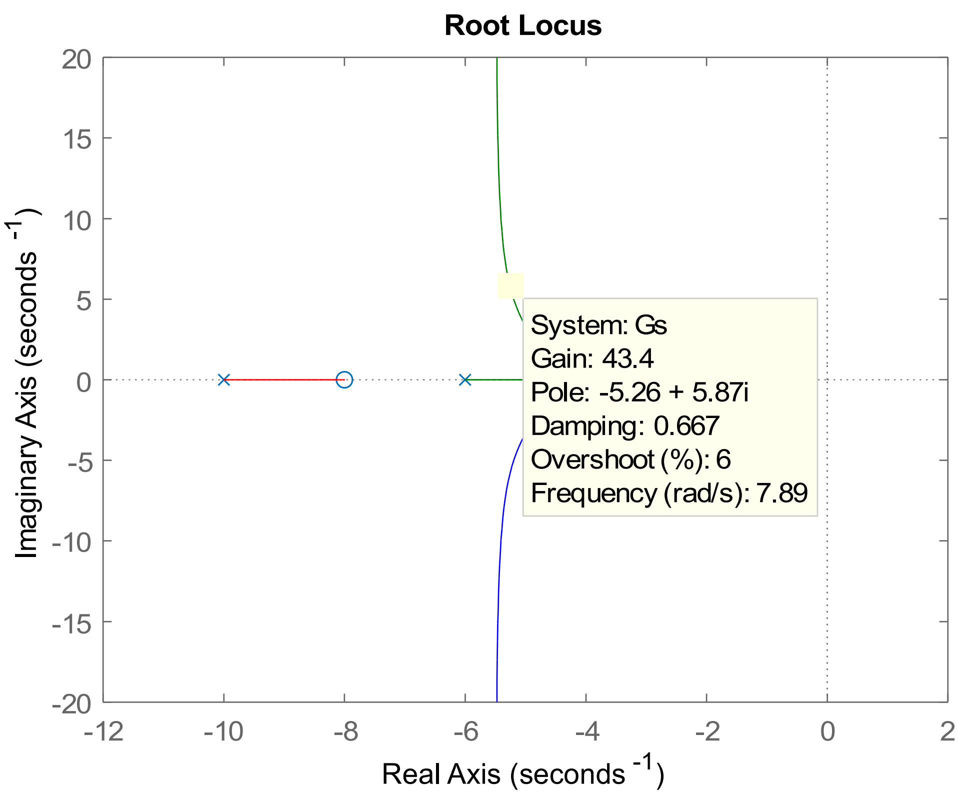
Untitled.m

```
clear all
close all
clc
Gs=zpk([-8],[-3 -6 -10],1)
rlocus(Gs)
```

Output:



```
Command Window

Gs =

        (s+8)
  ------------------
  (s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.
```



Task:

Editor - C:\Users\DELL\Untitled.m

Untitled.m

```
clear all
close all
clc
s= -8.157+15.92i;
Gs=1*(s+8)/((s+3)*(s+6)*(s+10));
phase=angle(Gs);
Theta=180/pi*phase
```

Output:

```
Theta =

  161.5037
```

Task:

Editor - C:\Users\DELL\Untitled.m

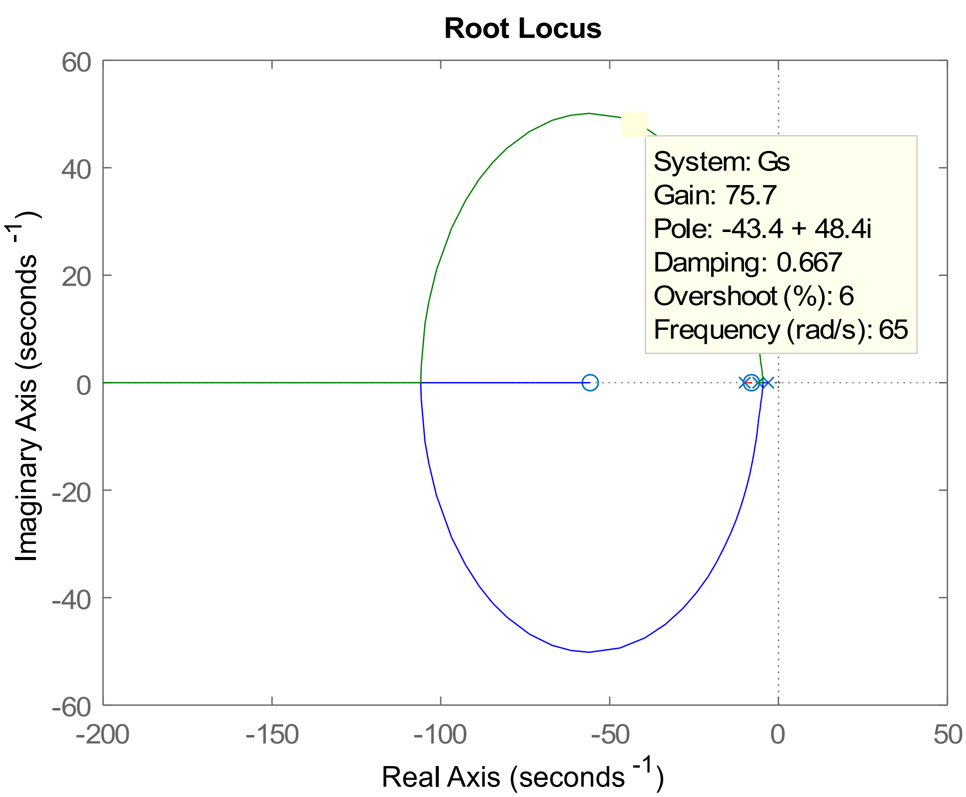
Untitled.m

```
clear all
close all
clc
Gs=zpk([-8 -55.75],[-3 -6 -10],1)
rlocus(Gs)
```

Output:

Command Window

```
Gs =

    (s+8) (s+55.75)
  ------------------
  (s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.
```



Task:

Editor - C:\Users\DELL\Untitled.m

Untitled.m

```
clear all
close all
clc
Gs=zpk([-8],[-3 -6 -10],122)
GPD=zpk([-8 -55.75],[-3 -6 -10],5.4)
Sp=feedback(Gs,1)
Spd=feedback(GPD,1)
figure(1)
impulse(Sp,1.5)
hold on
impulse(Spd,1.5)
figure(2)
step(Sp,1.5)
hold on
step(Spd,1.5)
```

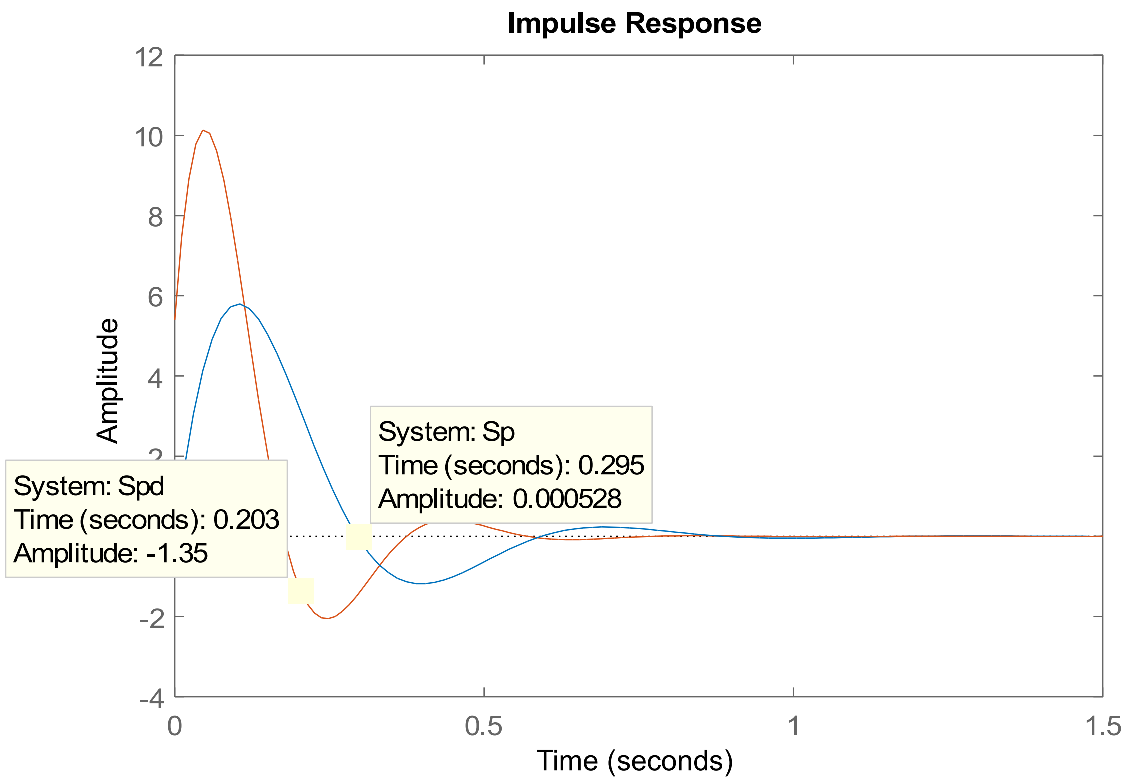
Output:

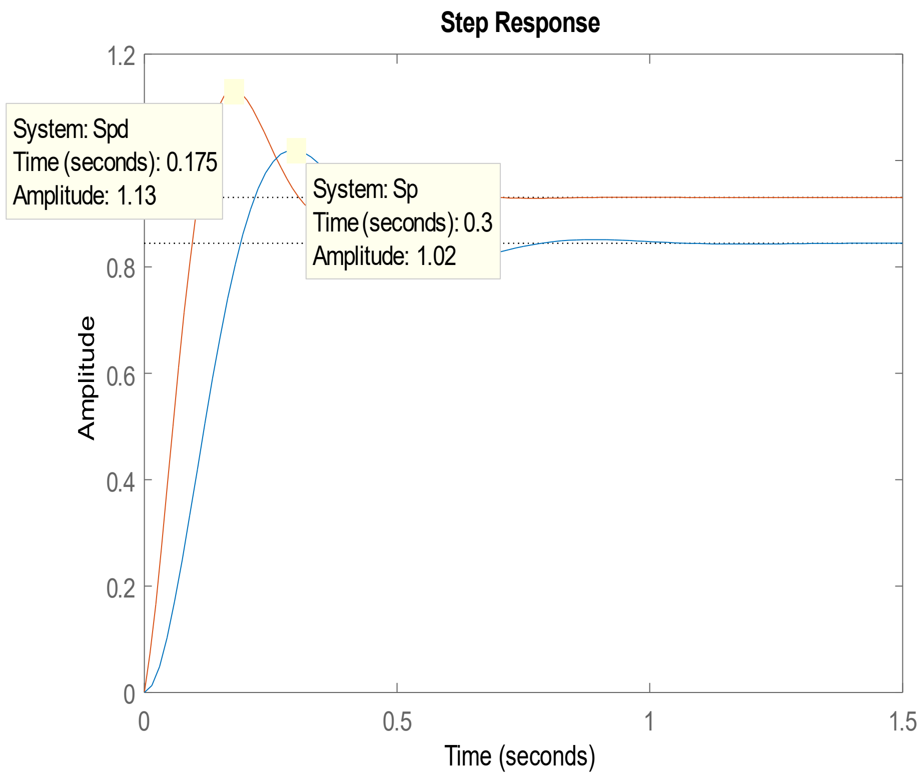
Gs =

```
    122 (s+8)
  ------------------
  (s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.


GPD =

  5.4 (s+8) (s+55.75)
  -------------------
  (s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.


Sp =

          122 (s+8)
  --------------------------------
  (s+8.168) (s^2 + 10.83s + 141.5)

Continuous-time zero/pole/gain model.


Spd =

      5.4 (s+55.75) (s+8)
  --------------------------------
```

(s+8.079) (s^2 + 16.32s + 320.4)

Continuous-time zero/pole/gain model.





Task:

Editor - C:\Users\DELL\Untitled.m

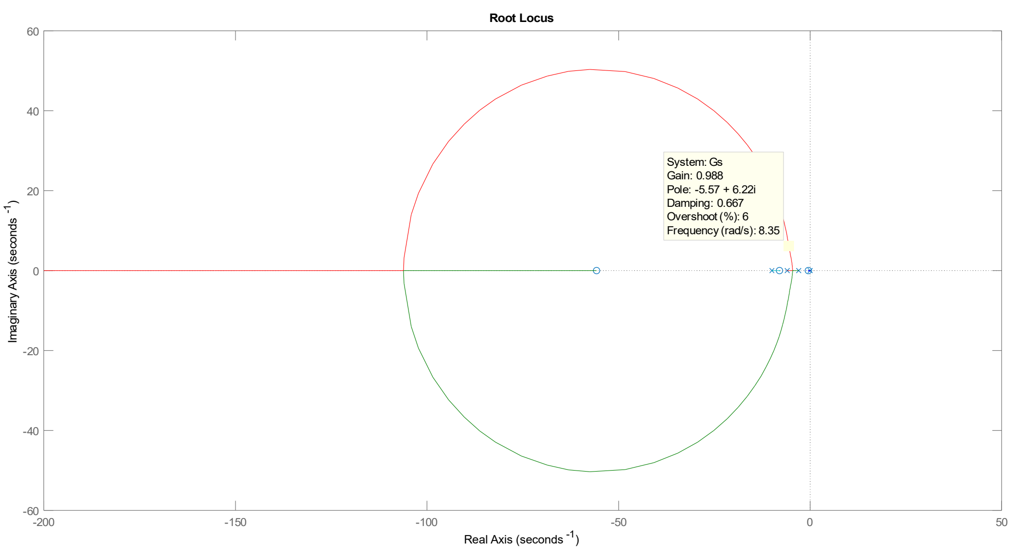
Untitled.m

```
clear all
close all
clc
Gs=zpk([-8 -55.75 -.5],[0 -3 -6 -10],1)
rlocus(Gs)
```

Output:

Command Window

```
Gs =

   (s+8) (s+55.75) (s+0.5)
  -------------------------
    s (s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.
```



Task:

Editor - C:\Users\DELL\Untitled.m

Untitled.m

```
clear all
close all
clc
Gs=zpk([-8],[-3 -6 -10],122)
GPD=zpk([-8 -55.75],[-3 -6 -10],5.4)
GPID=zpk([-8 -55.75 -.5],[0 -3 -6 -10],4.63)
Sp=feedback(Gs,1);
Spd=feedback(GPD,1);
Spid=feedback(GPID,1);
figure(1)
impulse(Sp)
hold on
impulse(Spd)
hold on
impulse(Spid)
figure(2)
step(Sp)
hold on
step(Spd)
hold on
step(Spid)
```

Output:

Gs =

```
    122 (s+8)
  ------------------
  (s+3) (s+6) (s+10)
```

Continuous-time zero/pole/gain model.

GPD =

5.4 (s+8) (s+55.75)

(s+3) (s+6) (s+10)

Continuous-time zero/pole/gain model.

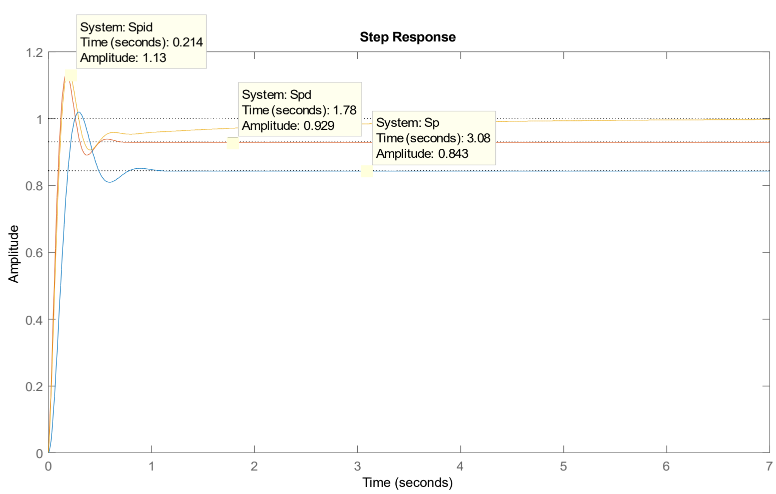
GPID =

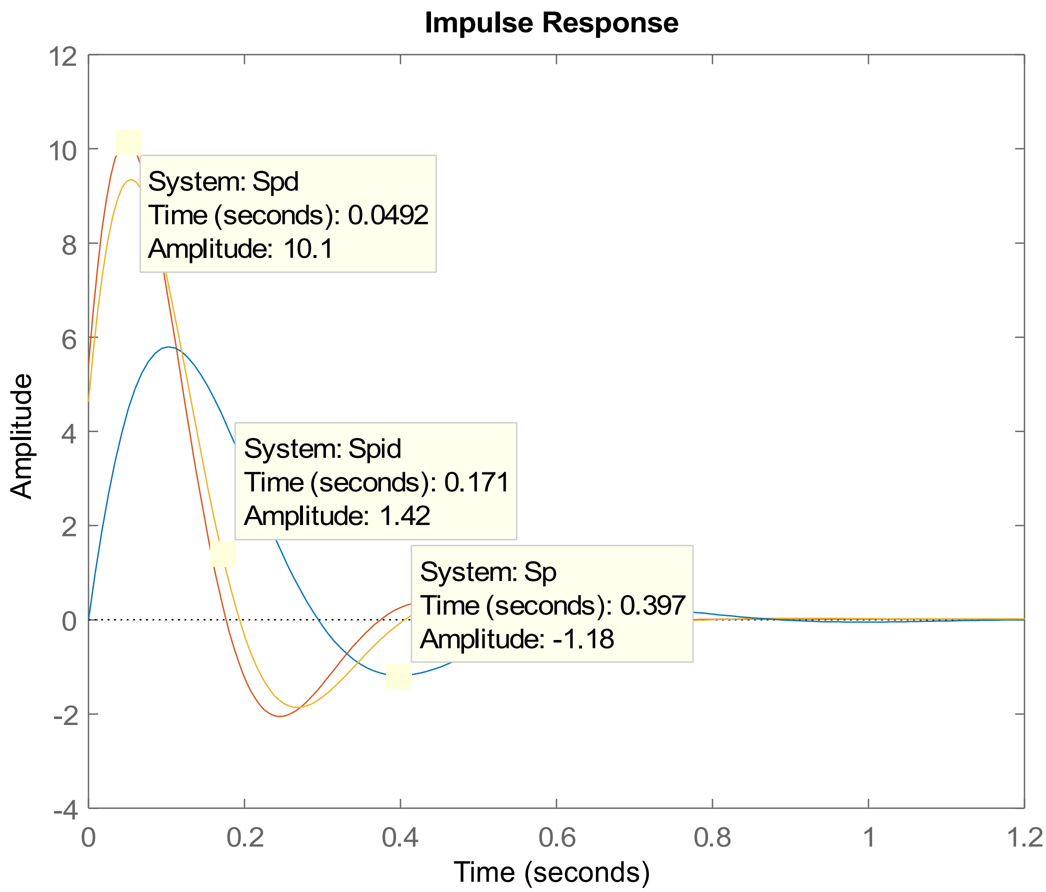
4.63 (s+8) (s+55.75) (s+0.5)

----

s (s+3) (s+6) (s+10)

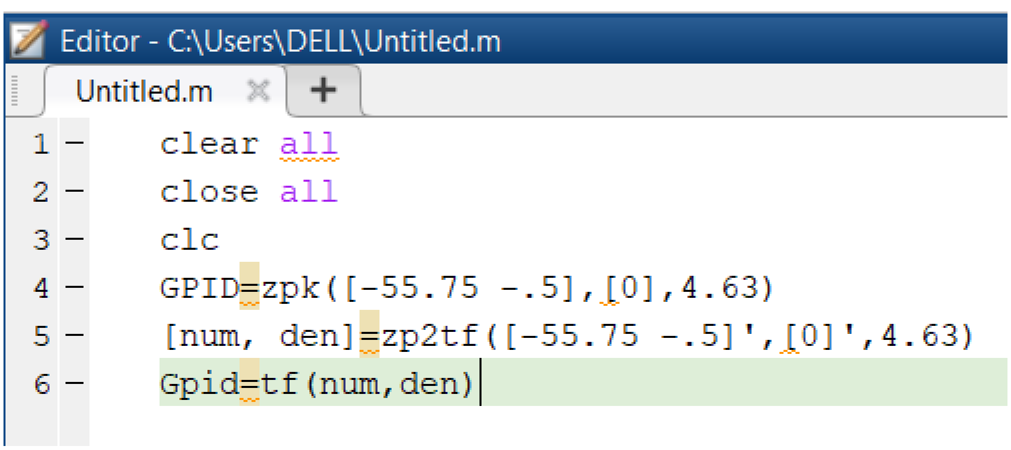
Continuous-time zero/pole/gain model.





Impulse Response
System: Spd
Time (seconds): 0.0492
Amplitude: 10.1
System: Spid
Time (seconds): 0.171
Amplitude: 1.42
System: Sp
Time (seconds): 0.397
Amplitude: -1.18
Amplitude
Time (seconds)
12
10
8
6
4
2
0
-2
-4
0
0.2
0.4
0.6
0.8
1
1.2

Task:



Editor - C:\Users\DELL\Untitled.m
Untitled.m
clear all
close all
clc
GPID=zpk([-55.75 -.5],[0],4.63)
[num, den]=zp2tf([-55.75 -.5]',[0]',4.63)
Gpid=tf(num,den)

Output:

```
Command Window

GPID =

  4.63 (s+55.75) (s+0.5)
  ----------------------
            s

Continuous-time zero/pole/gain model.


num =

    4.6300  260.4375  129.0613


den =

     1     0


Gpid =

  4.63 s^2 + 260.4 s + 129.1
  --------------------------
              s

Continuous-time transfer function.
```

Discussion:

This lab was absolute new to us. We learned and used few new functions of MATLAB. We used zpk() function which creates a special varible that can use in a rational expression to create a continuous time zero pole gain. We also used rlocus() function to find the root locus of a certain function. After creating any desired output we pointed out specific position and found the determinants and parameters. We held on the point of our respective roll number and found the parameters of that point. It was fun and innovative.

Conclusion:

We learned many new items from this experiment. We designed the PID controller system, we found out parameters and determinants of specific function and also specific location. We calculated the phase and steady-state error parameters of specific compensated systems.