Q2.

A. We have u(t)=0(input force=0) in this case and we must find step disturbance. (with magnitude 0.1)

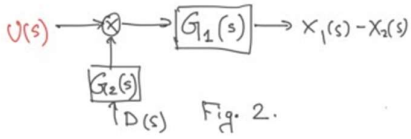


Fig. 2.

Since u(s)=0 we will have the transfer function d(t) to \ x1(t)-x2(t) i.e., the transfer function will be Gd.

Gd= (−2500*s^2)/(s^4+23.33*s^3+2791.666667*s^2+833.3*s+104166.6667)

- ⇨ X1(s)-X2(s)=0.1*Gd/s

So the step response is can be found with code,

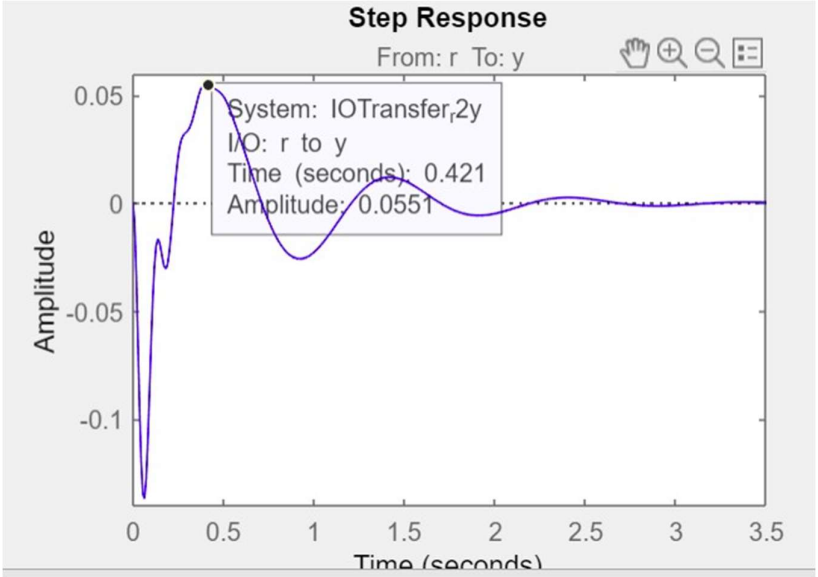
ilaplace((-250*s)/(s^4+23.33*s^3+2791.666667*s^2+8333.3*s+104166.6667).

Code

```
sys = tf([-250 0 0],[1 23.33333333 2791.666667 8333.3333333 104166.6667])
step(sys)
```

(-250 is taken because the magnitude of step disturbance is 0.1.

The figure for step response is as follows.



We see that the peak amplitude us 0.05. Here we cannot consider overshoot as the steady state response is 0. So, we consider absolute overshoot i.e., the overshoot here is 5.51%.

Code

sys = tf([-250 0 0],[1 23.33333333 2791.666667 8333.3333333 104166.6667])

S=stepinfo(sys)

S =

struct with fields:

RiseTime: 0

SettlingTime: 2.5967

SettlingMin: -0.1261

SettlingMax: 0.0503

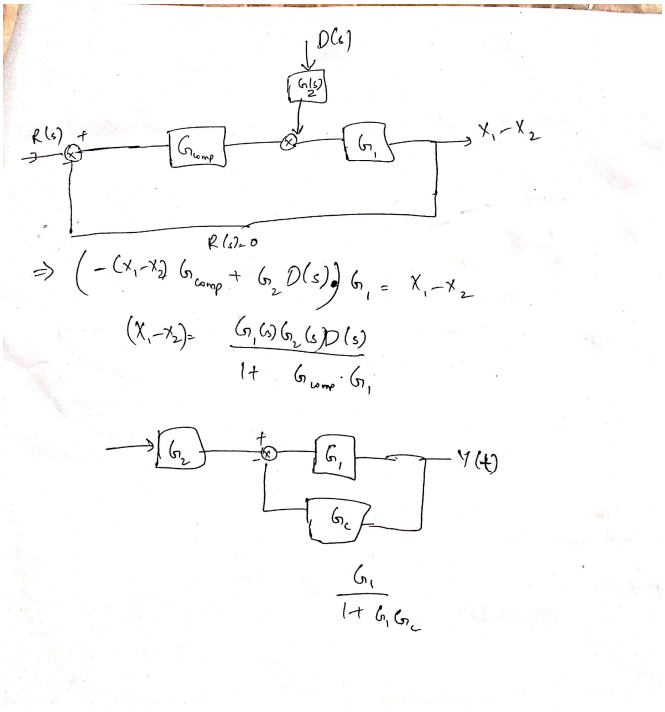
Peak: 0.1261

PeakTime: 0.0538

Overshoot=5.51%

Tsettling = 2.5967s

B.



Now we have to design a suitable compenstor (choose from P/PI/PD/PID/Lag/Lead/Lag-lead/Notch filter) that leads to a comfortable ride. We would like to check for step disturbance for all rides and the one which does not overshoot too much from the steady state and settles to steady state quickly is ideally the most opted choice. We choose a step disturbance of input 0.1 magnitude.

So let us start with the Proportional compensator(P).

1. Proportional compensator(P)

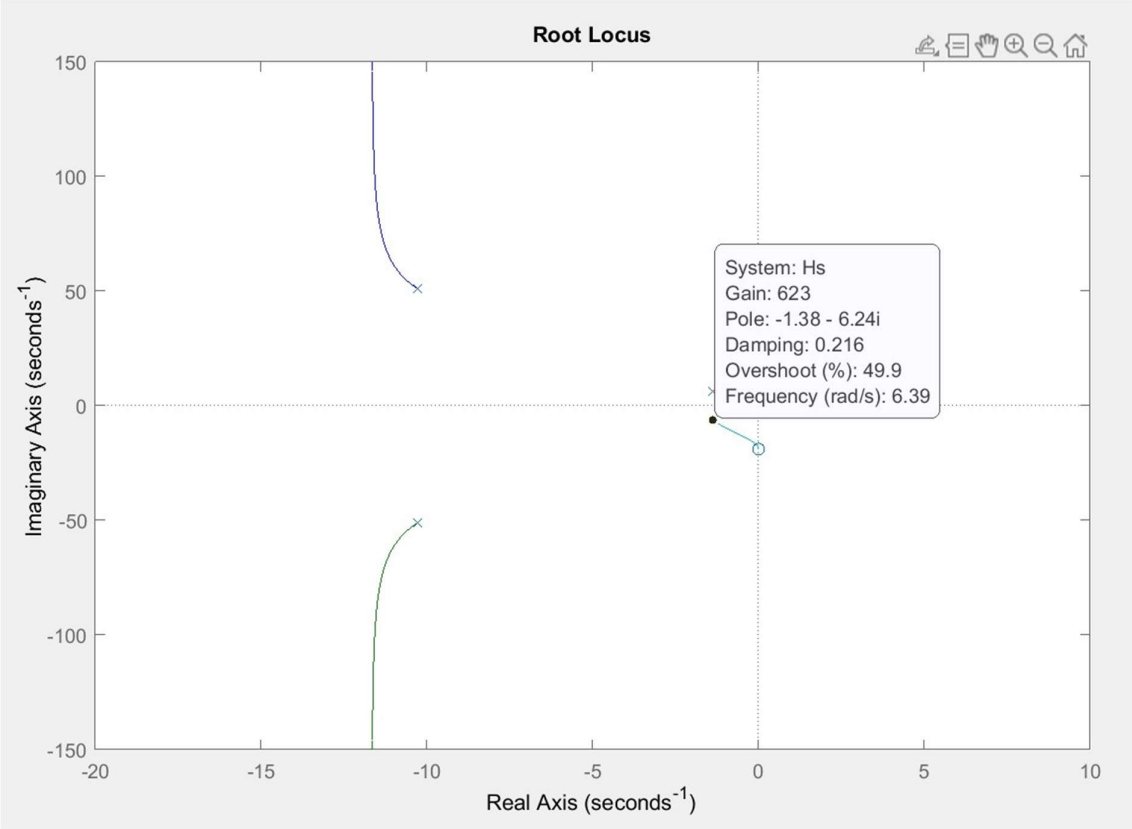
For P compensator we use Gc=K1

So, for a gain K we take root locus of the transfer function G1(as it is inside the feedback loop).

G1=(0.02916666667*s^2+10.41666666660714)/(s^4+23.333333*s^3+2791.6667*s^2+8333.333*s+104166.66777)

Code

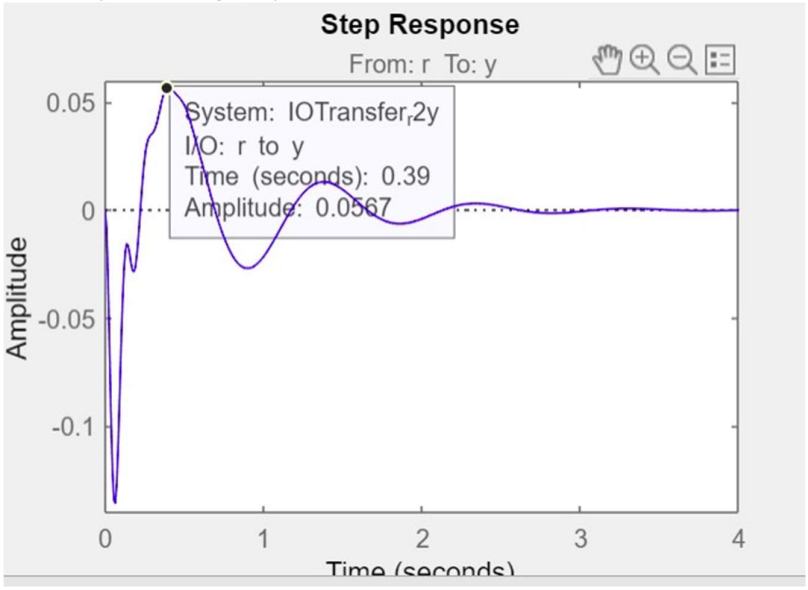
```
num = [0.02916666667 0 10.41666666660714]
den = [1 23.33333333 2791.666667 8333.3333333 104166.6667]
Hs=tf(num,den)
rlocus(Hs)
```



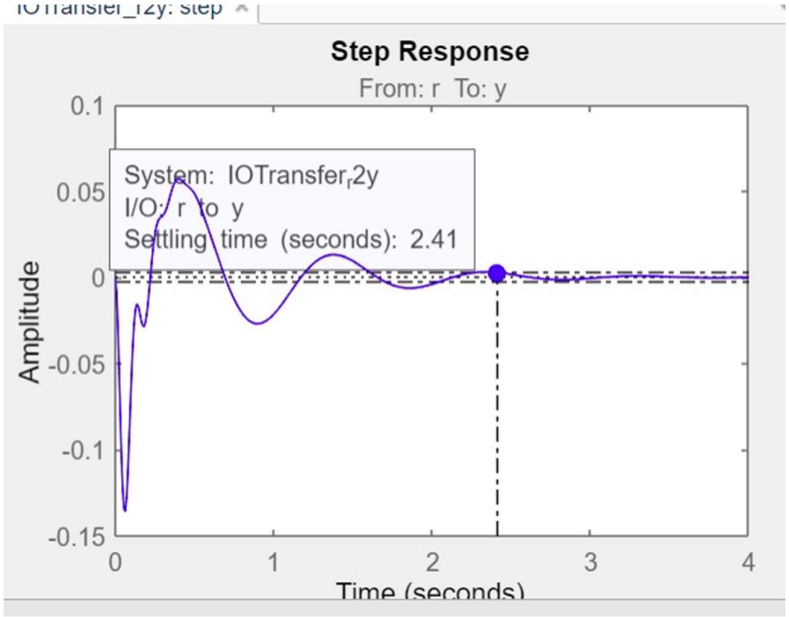
We find step response,

Code

```
s = tf('s');
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+623*(0.0291666666
7*s^2+10.4166666667));
controlSystemDesigner(plant)
```



We can see overshoot is 5.67% in the best case for P compensator, which actually is not better than the normal one without step response.



Settling time is 2.41 seconds.

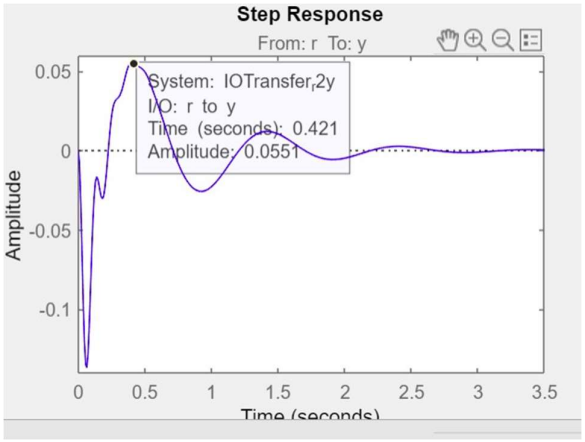
2. PI compensator

For PI compensator we use Gc=(K1s+K2)/s

We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and also for gain.

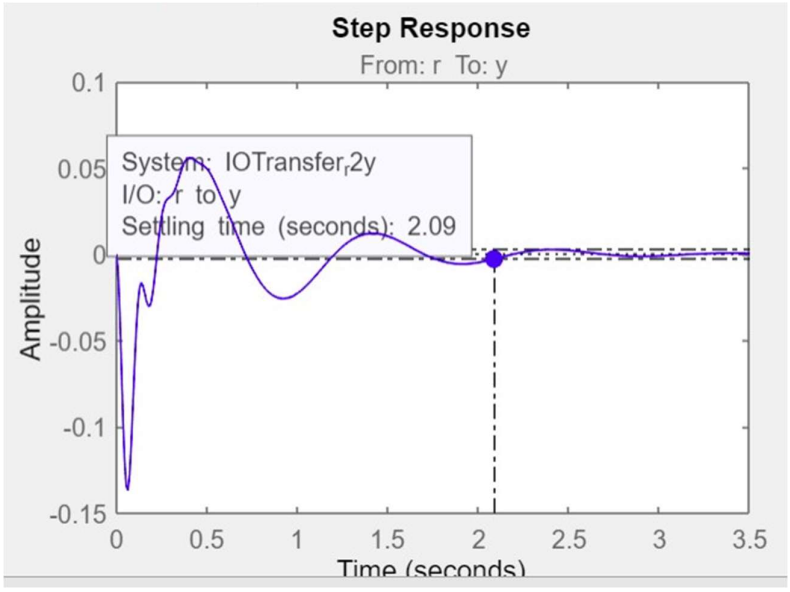
Code

```
s = tf('s');
plant = (-250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+7.84e-11*(s+0.1)*(0.02916666667*s^2+10.4166666667)/s);
controlSystemDesigner(plant)
```



We can see that the best case in this is when it almost approximates to the open loop transfer function without the compensator. Hence PI compensator is not of much use in this case.

Overshoot=5.51%



Settling time is 2.09s

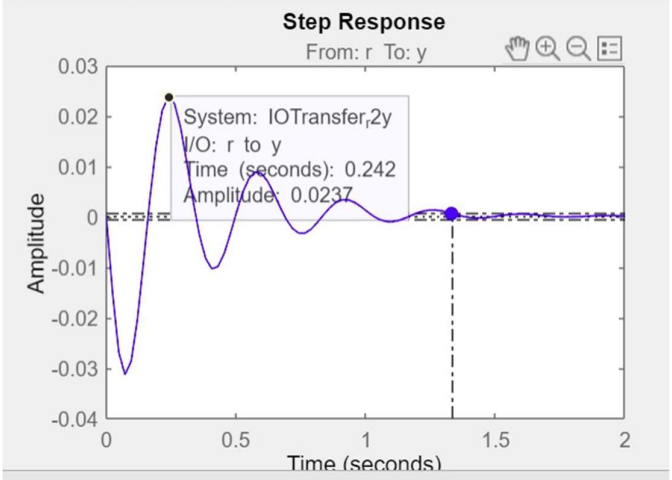
PD Compensator

For PD compensator we use Gc=K1s+K2

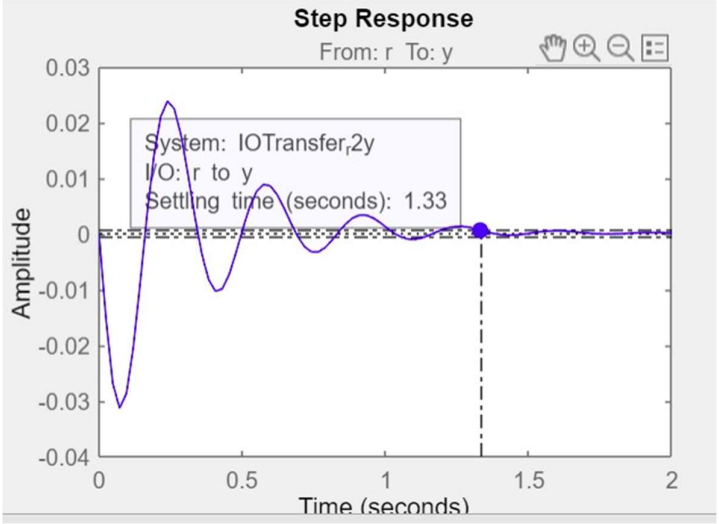
We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and for gain.

Code

```
s = tf('s');
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+10000*(s+1)*(0.029
16666667*s^2+10.4166666667));
controlSystemDesigner(plant)
```



Overshoot here is 2.37%



Settling time is 1.33s.

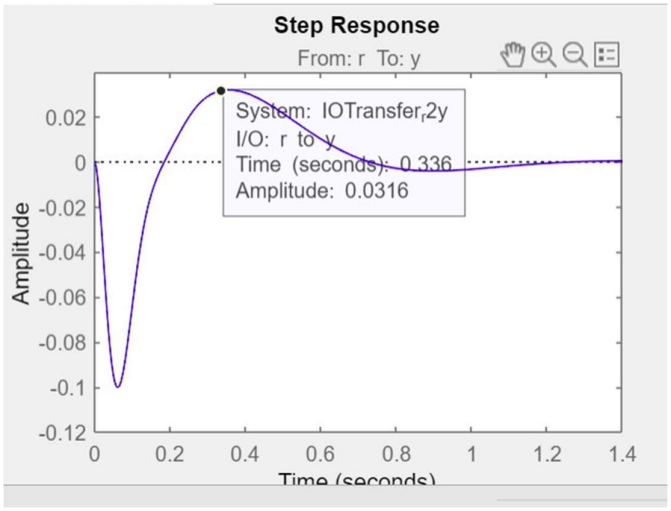
PID compensator

We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and for gain.

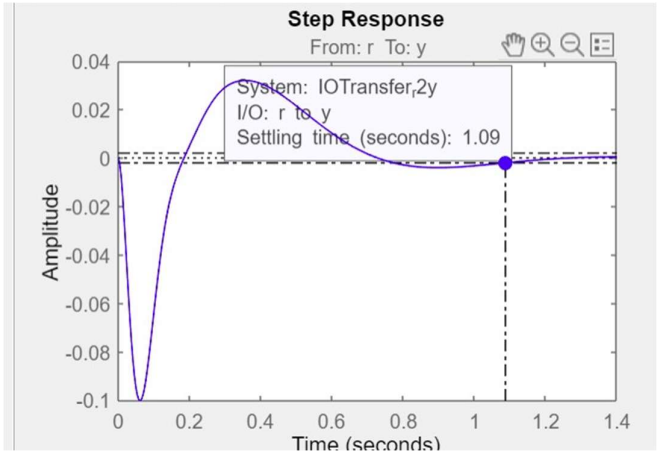
For PID compensator we use Gc=K1s^2+K2s+K3

Code

```
s = tf('s');
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+(1000*s^2+50*s+1)
*(0.02916666667*s^2+10.4166666667)/s);
controlSystemDesigner(plant)
```



Overshoot here is 3.16%



Settling time is 1.09s

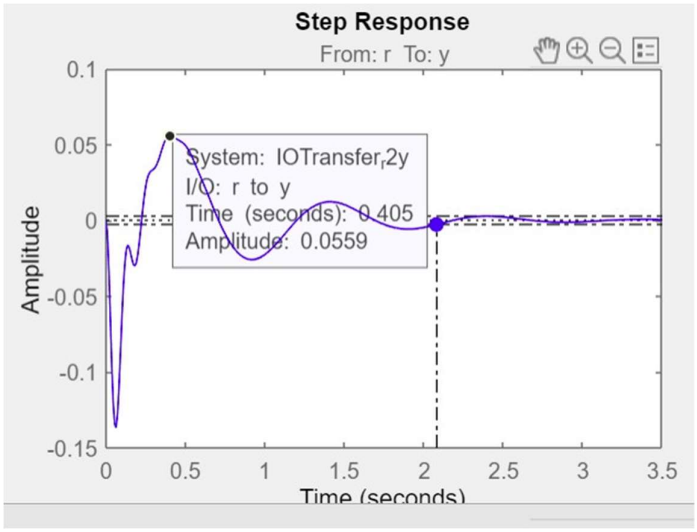
Lead Compensator

We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and for gain.

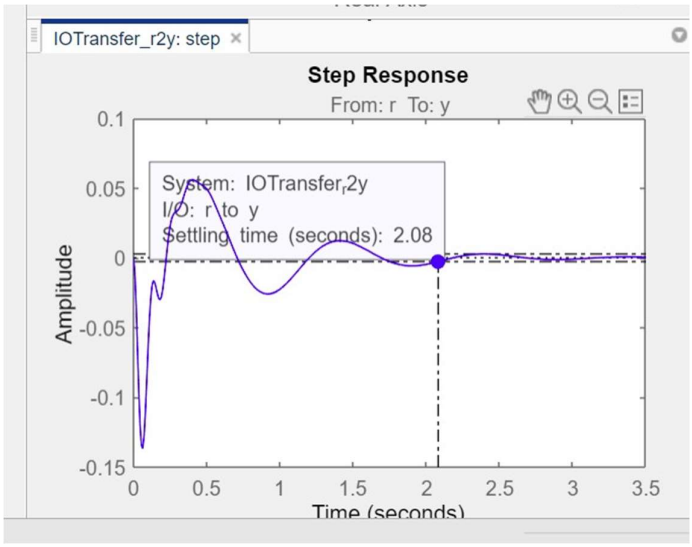
For Lead compensator we use Gc=(K1s+K2)/(s+K3) [(K(s+zc)/(s+zp))]

Code

```
s = tf('s')
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+(0.02916666667*s^
2+10.4166666667)*(100*s+5000)/(s+5000000));
controlSystemDesigner(plant)
```



Overshoot is 5.59%



Settling time is 2.08s

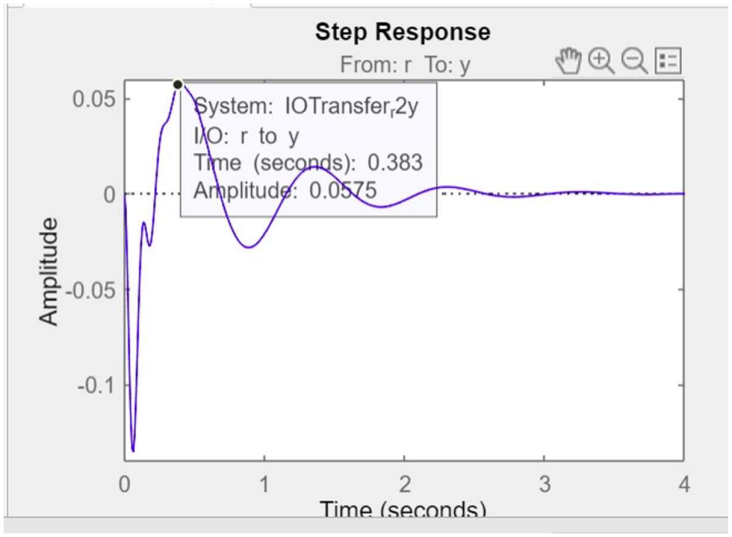
Lag Compensator

We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and for gain.

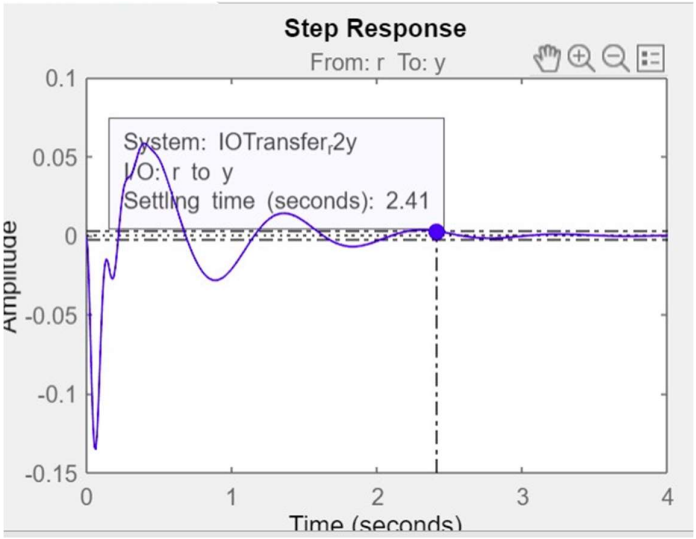
For Lag compensator we use Gc=(K1s+K2)/(s+K3) (K(s+zc)/(s+zp))

Code

```
s = tf('s')
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+(0.02916666667*s^
2+10.4166666667)*(1000*s+555)/(s+0.005));
controlSystemDesigner(plant)
```



Overshoot is 5.75%



Settling time is 2.41s

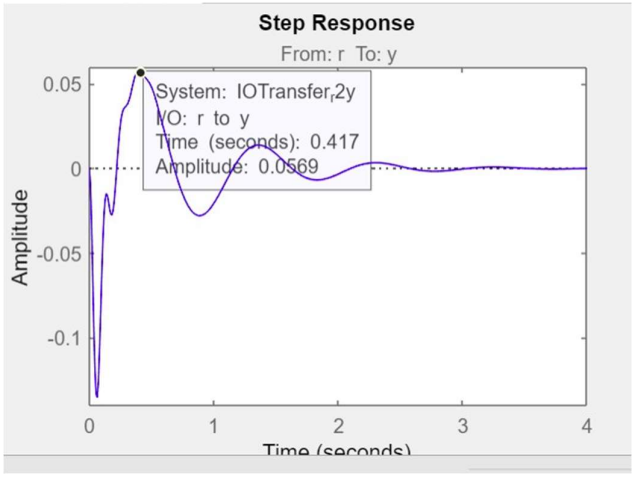
Lag-Lead Compensator

We find the best case of overshoot in this case and find gain. We set random values to the roots that is to be added and for gain.

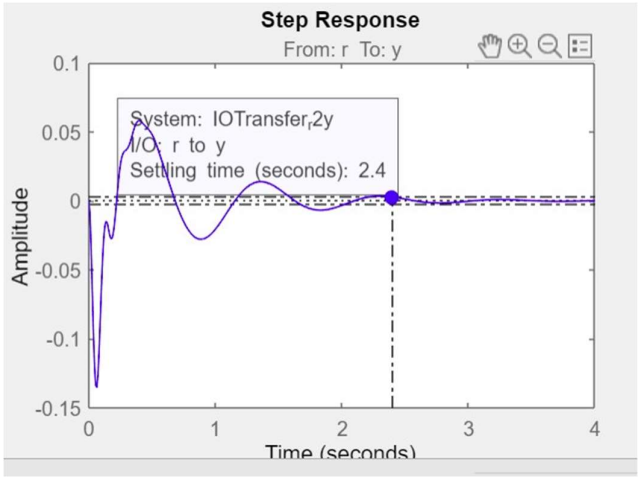
For Lag-Lead compensator we use Gc=(K1s^2+K2s+K3)/(s^2+K4s+K5)

Code

```
s = tf('s')
plant = (-
250*s^2)/(s^4+23.33333333*s^3+2791.666667*s^2+8333.333*s+104166.6667+(0.02916666667*s^
2+10.4166666667)*(1000*s^2+200*s+555)/(s^2+0.005*s+0.5));
controlSystemDesigner(plant)
```



Overshoot is 5.69%



Settling time is 2.4s

Conclusion

We had simulated normal step response when control force=0. We compare it the ones that we add compensators. We aim to decrease overshoot and settling time. For all compensators that we have designed are for the requirements. We had used random numerical for all of them. We can see that from all the compensators that we have simulated PD is probably the best compensator to reach the requirements. Also, in PD compensator overshoot is more than PID but the good factor is that PID has a faster settling time than PD compensator. We ideally need systems have less overshoot and less settling time. But we cannot compare the cases like PD, PID without the help of physical tests. We must apply both to physical test and compare the results of the person and find which is the one with a comfortable ride.