



The Hashemite University
Faculty of Engineering

Control system

Control's project


Name:	Major:	ID
Osama Ziad Abu-Jweed	Computer Engineering	1632968
Osama Yehya Johar	Computer Engineering	1632971
Mu'men Fayez Abu-Zeneh	Mechatronics Engineering	1636716
Israa Ahmad Fedda	Computer Engineering	1632972
Bothaina Anwar Zakarneh	Biomedical Engineering	1632876
Sameeha Almobiadeen	Biomedical Engineering	1636920
Bahaa Tayseer Al-Khader	Biomedical Engineering	1632879

Introduction:

We suggested a suspension system as system that we modeled and simulated it :

PURPOSE OF SUSPENSION SYSTEM

- Supports the weight.
- Provides a smooth ride.
- Allows rapid cornering without extreme body roll.
- Keeps tires in firm contact with the road.
- Allows front wheels to turn side-to-side for steering.
- Works with the steering system to keep the wheels in correct alignment.
- Isolate passenger and cargo from vibration and shock

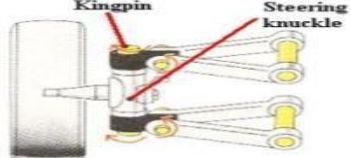



Suspension System

Basic Parts:

Control Arm:- movable lever that fastens the steering knuckle to the vehicle's body or frame.

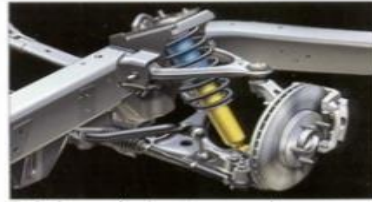
Steering Knuckle:- provides a spindle or bearing support for the wheel hub, bearings and wheel assembly.



Suspension System

Basic Parts:

Ball Joints:— swivel joints that allow control arm and steering knuckle to move up and down and side to side.



Today's complex import suspension systems aren't tolerant of excessive wear.

Springs:— supports the weight of the vehicle; permits the control arm and Wheel to move up and down.



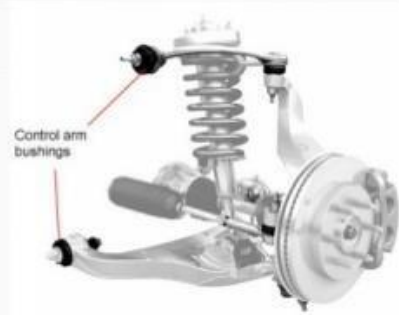
Suspension System

Basic Parts:

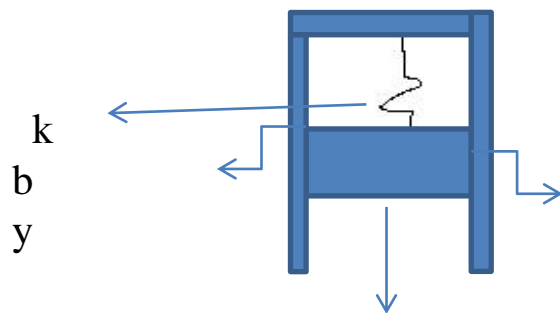
Shock absorbers or dampeners:— keeps the suspension from continuing to bounce after spring compression and extension.



Control arm bushing :— sleeves that allows the control arm to swing up and down on the frame.



Modeling analysis :



$$F(t)$$
$$\sum f = My''$$

$$F(t) - ky - by' = My''$$

$$My'' + by' + ky = f(t)$$

Take a lapalcian to find transfer function :

$$Ms^2y(s) + bsy(s) + ky(s) = f(s)$$

We supposed that : input= $y(s)$, output= $f(s)$

$$G(s) = y(s)/f(s) = 1/ms^2 + bs + k$$

And we supposed $M=3$, $b=1.5$, $k=1$

We have a unit feedback signal $H(s)=1$

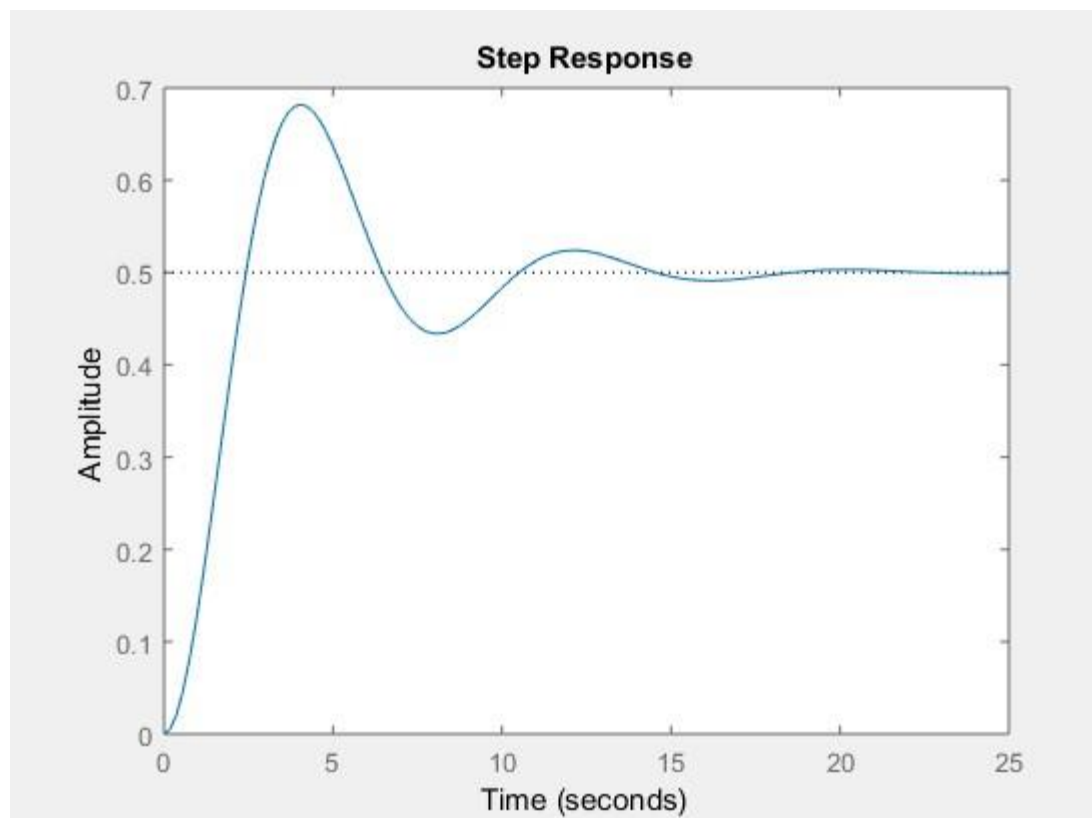
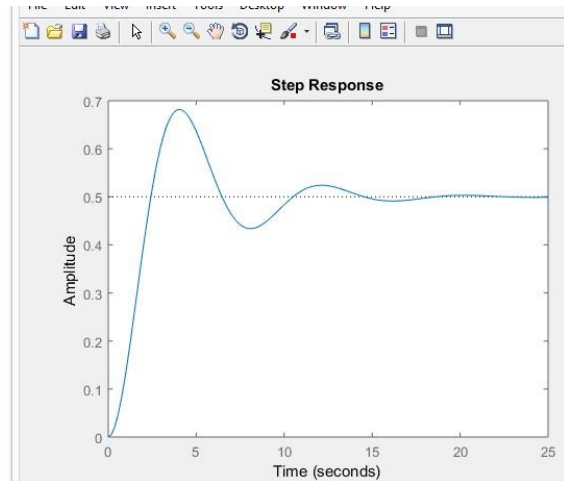
Simulation part :

after we enter the transfer function to MatLab to get response step of the transfer function , we have got

the next response :

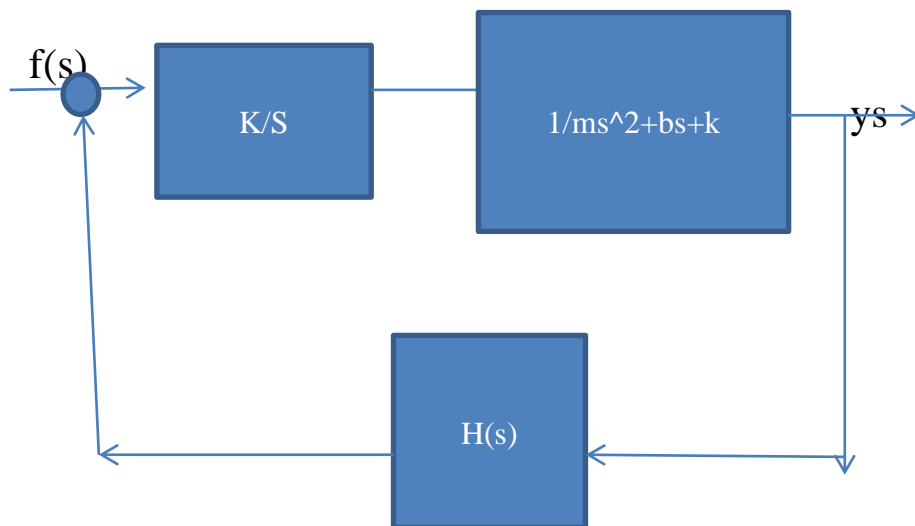
on the work space :

```
1 clear all
2 clc
3
4 m=3;
5 b=1.5;
6 k=1;
7
8 syms s
9
10 Gnum=1;
11 Gden=sym2poly(m*s^2+b*s+k);
12
13
14 G=tf(Gnum,Gden)
15
16 H=1;
17
18 T=feedback(G,H)
19
20 step(T)
21
22
23
24
```



- We can see that the response is not stable as good as we hope so we should find a control to make much more stable.

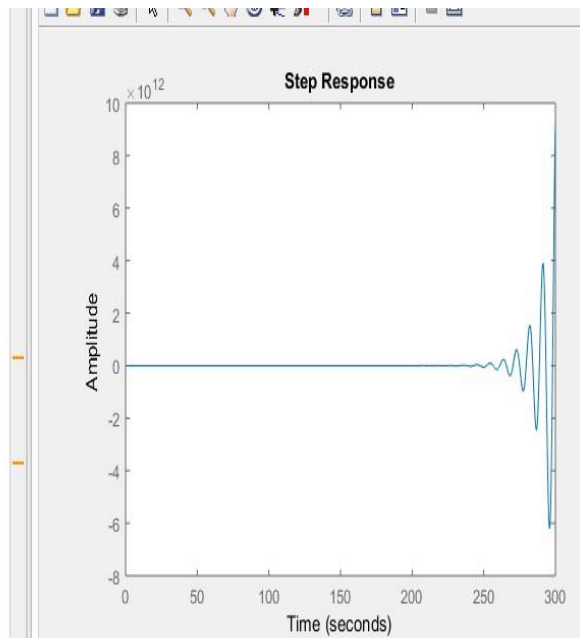
□ Selected controller :



```

2 - clc
3
4 - m=3;
5 - b=1.5;
6 - k=1;
7 - K=1;
8
9 - syms s
10
11 - Gnum=1;
12 - Gden=sym2poly(m*s^3+b*s^2+k*s);
13
14
15 - G=tf(Gnum,Gden)
16
17 - H=1;
18
19 - T=feedback(G,H)
20
21 - step(T)
22
23
24
25

```

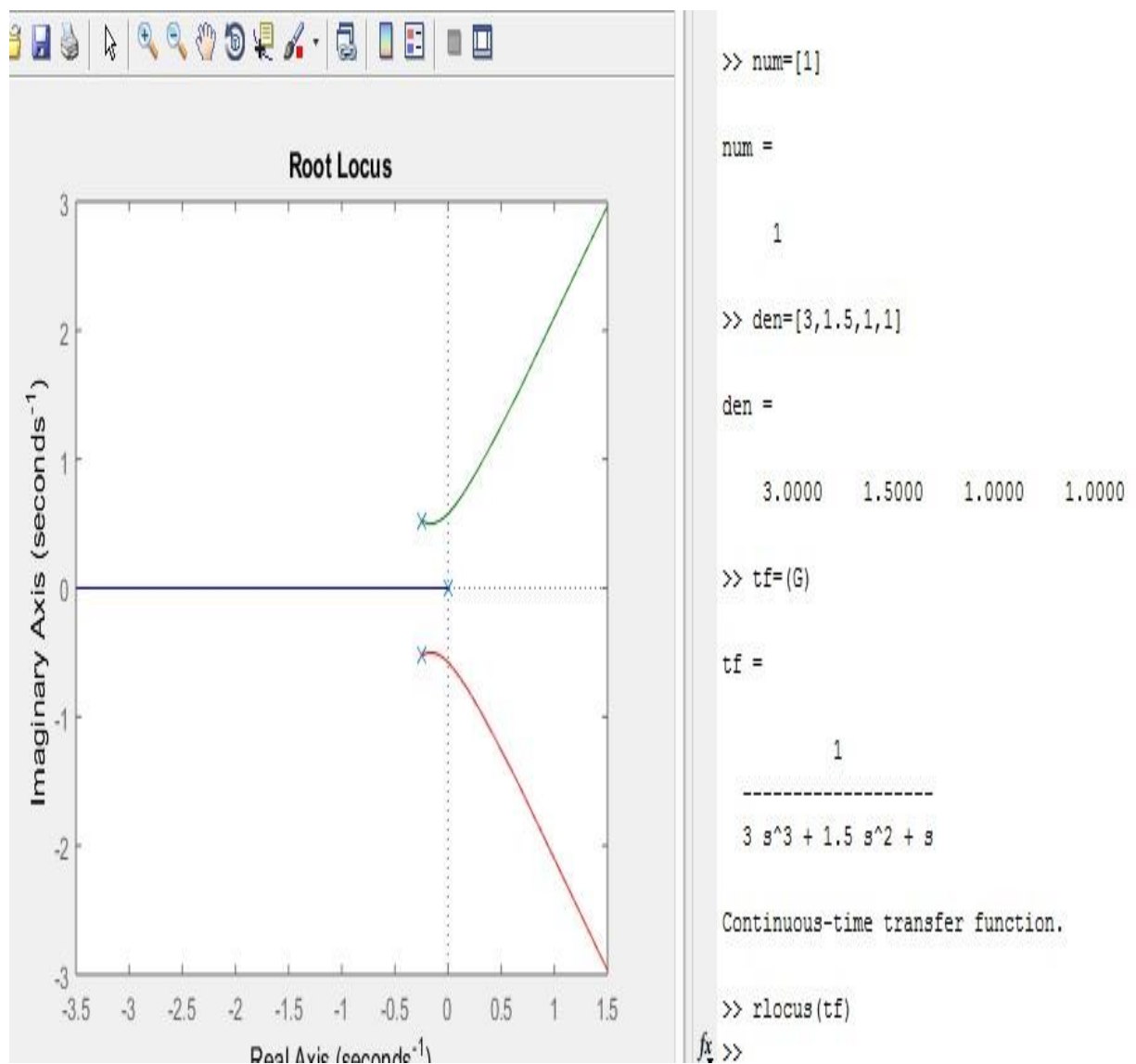


-So the control will be (K/s)

gain $K = 0.0175$

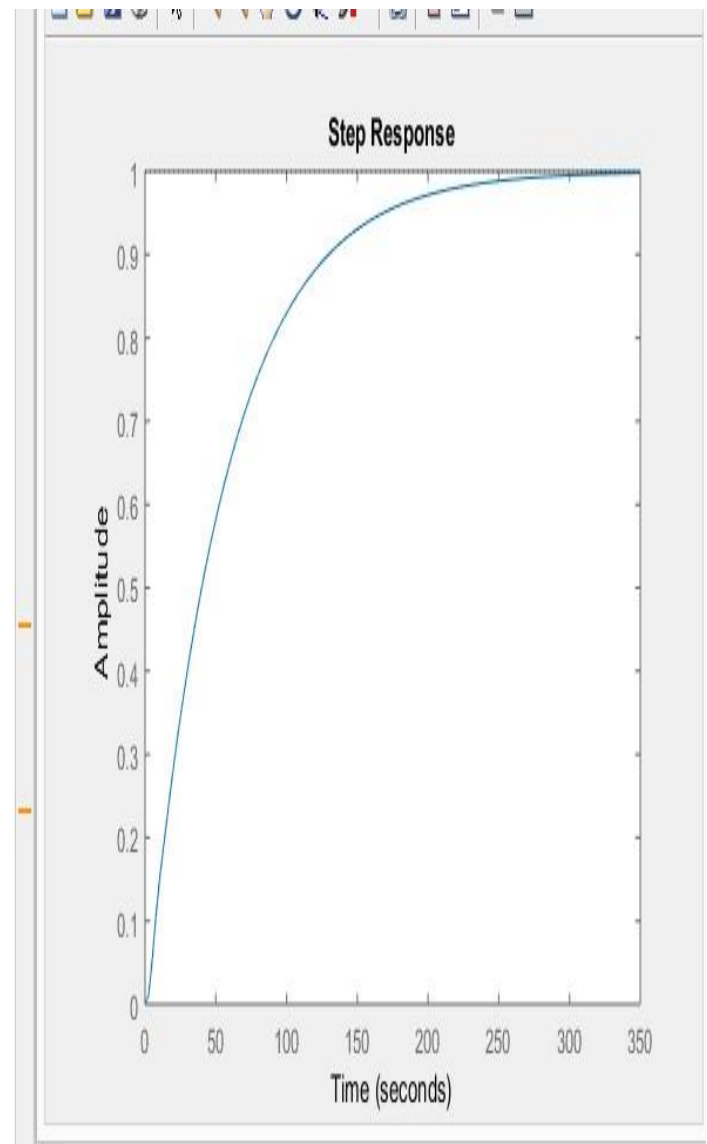
-We found gain(K) by using root locus on MatLab
, by search about the values that they have the
lowest overshoot .

Let us watch the next graph :



Now we have got the new response :

```
2 - clc
3
4 - m=3;
5 - b=1.5;
6 - k=1;
7 - K=0.0175;
8
9 - syms s
10
11 - Gnum=1*K;
12 - Gden=sym2poly(m*s^3+b*s^2+k*s);
13
14
15 - G=tf(Gnum,Gden)
16
17 - H=1;
18
19 - T=feedback(G,H)
20
21 - step(T)
22
23
24
25
```



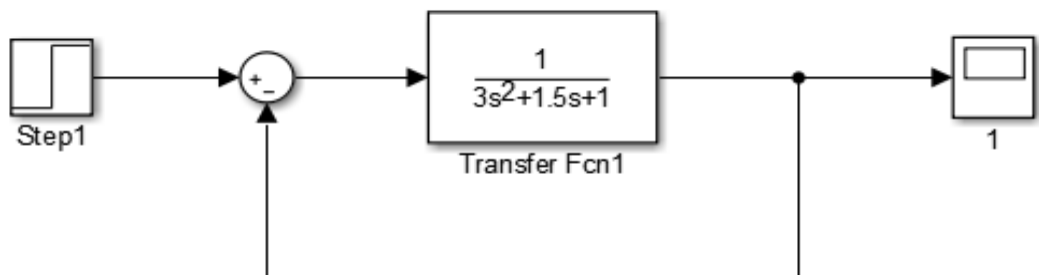
Discussion :

- We can see that in the first paragraph the response was not stable enough .
- So we used a control to help us to have a much more stable response .
- The control was (gain (K)/s).
- Gain K was found by using locus root .
- $K = 0.0175$
- After we enter the controller to the system , we have got a much more stable response .
- We can see that before we entered the control , the time is taking to reach to the steady state value is shorter than after entered the control .
- When we entered the control (K/s) when $K = 1$, there was no stable response (worst response stability) , so we used root locus to find a better gain K which it has the lowest overshoot .

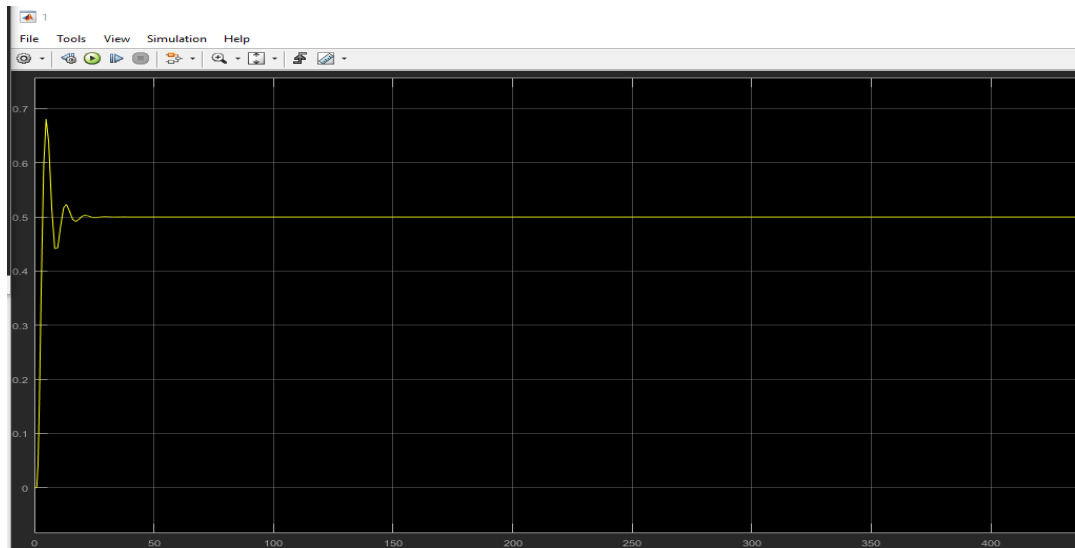
Simulink part :

After we enter the transfer function to MatLab to get response step of the transfer function , we have got the next response :

on the work space :

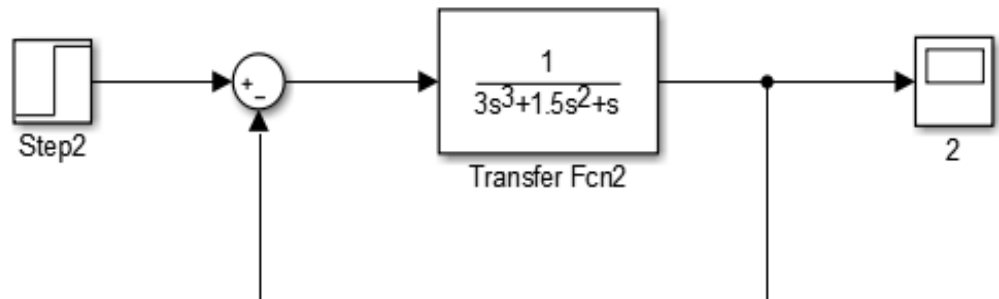


The scope 1 after 550 :



We can see that the response is not stable so we should find a control to make much more stable.

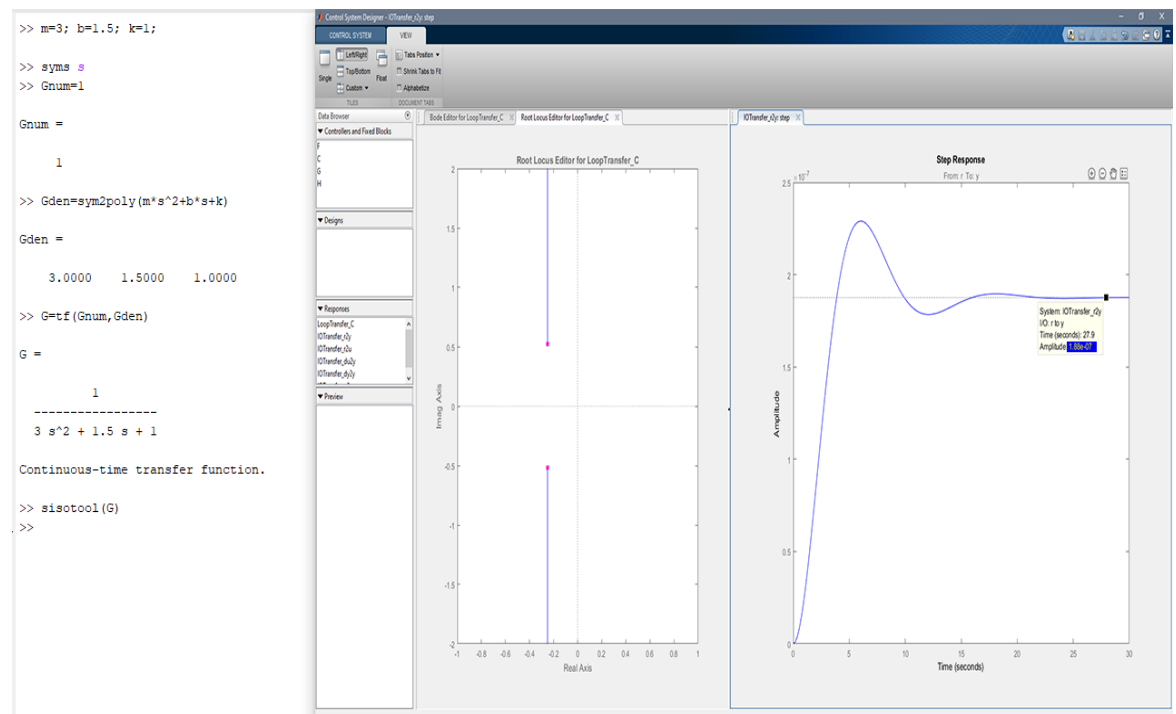
After selected controller :



The scope 2 after 550 :



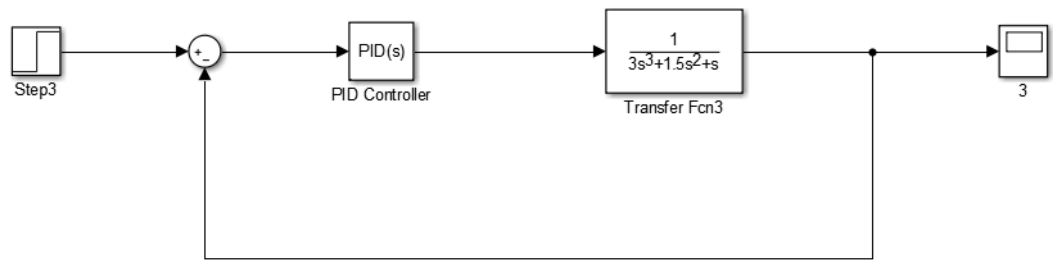
So we must control the system need some value from root locus as



We use sisotool in MatLab to show value we need as show the value we read it's : 1.8759e-7 we take this value and put that in PID in Integral :

The figure shows the 'Block Parameters: PID Controller' dialog box. The 'Controller' is set to 'PID' and the 'Form' is set to 'Parallel'. The 'Time domain' is set to 'Continuous-time'. The 'Main' tab is selected, showing the 'Controller parameters' section. The 'Source' is set to 'internal'. The 'Proportional (P)' gain is 0.05204165, the 'Integral (I)' gain is 1.8759e-07 (highlighted in yellow), the 'Derivative (D)' gain is 0.05204165, and the 'Filter coefficient (N)' is 100. The 'Tune...' button is visible. The 'Initial conditions' section shows 'Source' as 'internal', 'Integrator' as 0, and 'Filter' as 0. The 'External reset' is set to 'none'. The 'Ignore reset when linearizing' checkbox is unchecked, and the 'Enable zero-crossing detection' checkbox is checked. The 'OK' button is highlighted.

The system shape will be :



The scope 3 after 550 :

