Department of Electronics and Communication Engineering
PES University

CONTROL SYSTEMS

(UE21EC241B)

PROJECT REPORT

Topic: Thrust Vector Control of a Ballistic Missile

Team members:

Sujal L (PES1UG21EC300)

Sumanth R (PES1UG21EC301)

Sumukh S (PES1UG21EC302)

Semester 4, 'F' Section

CONTENTS

AIM OF THE PROJECT

Ballistic missiles are a type of missile that follow a ballistic trajectory, meaning that they are initially propelled into the atmosphere and then follow a free-falling trajectory towards their target. The key advantages of the ballistic missile are having long range and carrying nuclear warheads.

The thrust vector of a missile refers to the direction of the force generated by the missile's propulsion system. In other words, it is the direction in which the missile's engines push the missile forward.

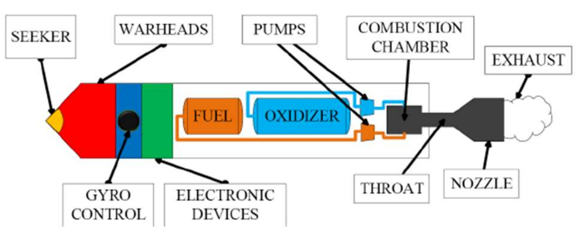
Nozzle angle, on the other hand, refers to the angle of the engine nozzle with respect to the missile's longitudinal axis. The nozzle angle can be fixed or movable, depending on the design of the missile. A movable nozzle can be used to achieve thrust vector control by deflecting the exhaust flow and changing the direction of the thrust vector.

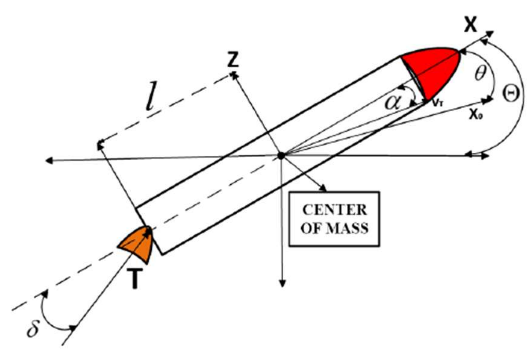
The relationship between thrust vector and angle of the nozzle is vital for controlling the movements of ballistic missiles.

Thrust vector control (TVC) is a technology used in some missiles and rockets to control the direction of the thrust vector. Thrust vector control using nozzle angle is a technique used in some missile designs to control the direction of the missile's thrust vector by adjusting the angle of the nozzle.

Here, we design a PID controller system to control the relationship between thrust angle and nozzle angle. The aim of missile control system which is being designed is to protect the relationship between nozzle angle and thrust angle on ballistic missiles. Any difference in nozzle angle must directly affect the angle of thrust vector, and this is critical in the missile maintaining a zero angle of approach to prevent being hit by an external entity.

Given below are the figures of parts of a rocket engine, and the missile parameters:





After assuming many of the values of the above missile parameters, a transfer function of the thrust angle and the nozzle angle is obtained.

DETAILS OF THE OPEN LOOP TRANSFER FUNCTION

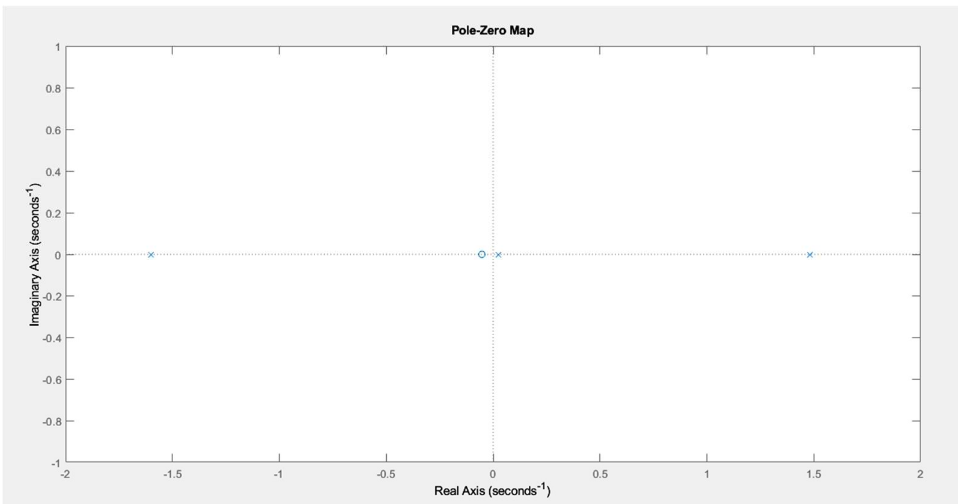
$$G(s) = \frac{\theta(s)}{\delta(s)} = \frac{-7.21(s+0.0526)}{(s+1.6)(s-1.48)(s-0.023)}$$

$\theta(s)$ is the thrust angle, $\delta(s)$ is the nozzle angle.

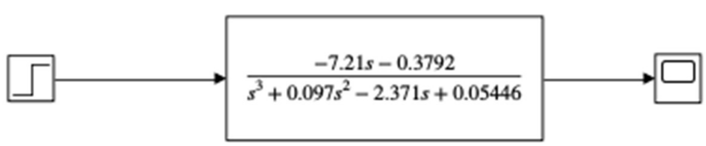
Poles: -1.6, +1.48, +0.023

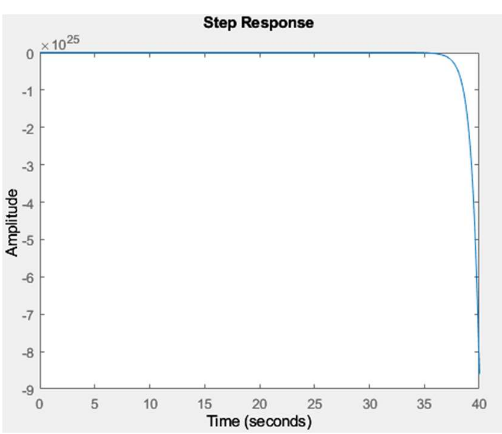
Zeroes: -0.0526

Pole zero map for the open loop system:

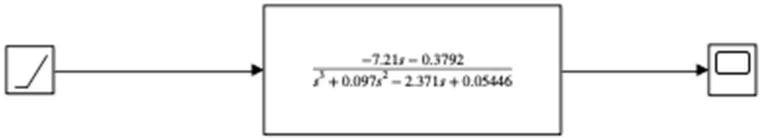


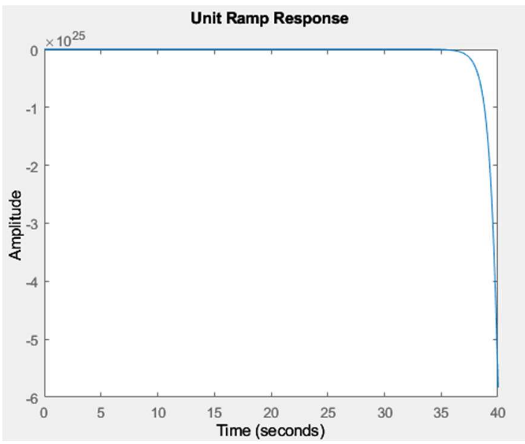
Open loop response for unit step input:





Open loop response for unit ramp input:



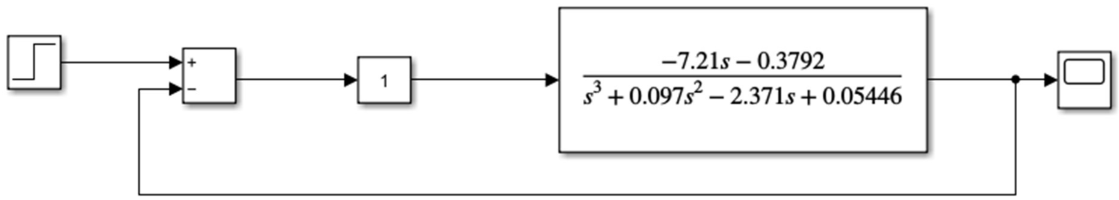


Due to the absence of complex poles, there are no oscillations present in the unit step and unit ramp responses. The responses are unbounded.

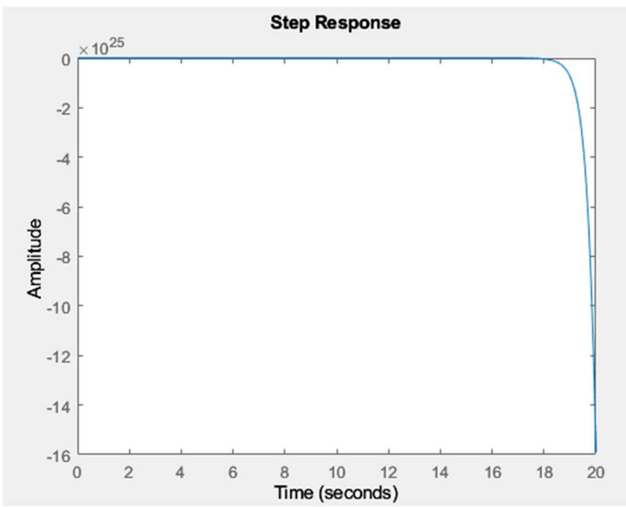
The open loop transfer function is unstable, due to the presence of 2 right hand plane poles in the s domain. The output of the system tends to -infinity. Thus, we cannot obtain time domain specifications.

UNITY FEEDBACK SYSTEM

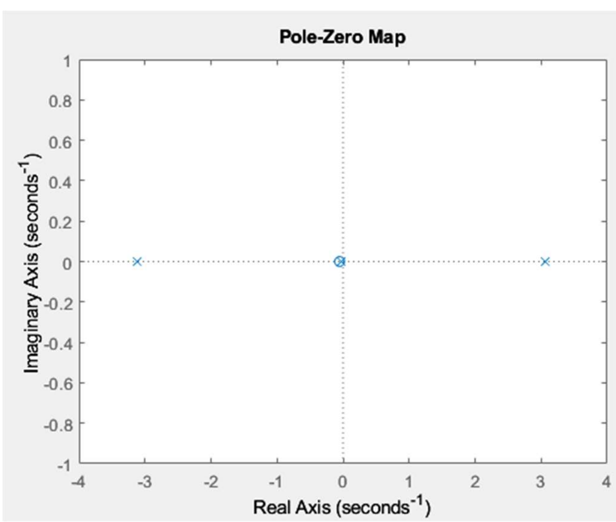
Let us now consider the plant transfer function to be part of a unity negative feedback system. Also consider a gain K kept in cascade to the given system. The value of K is initially kept as 1.



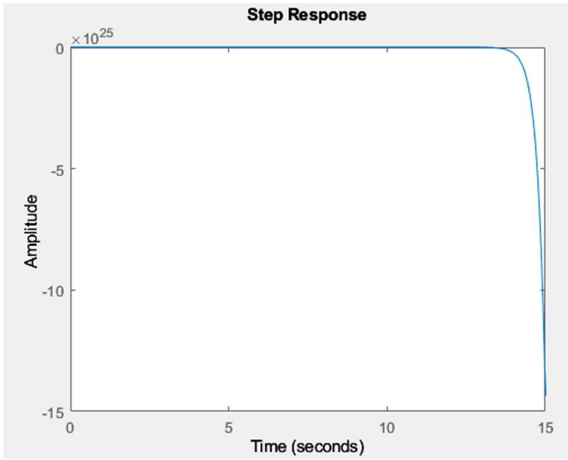
Unit step response of close loop system for k=1:



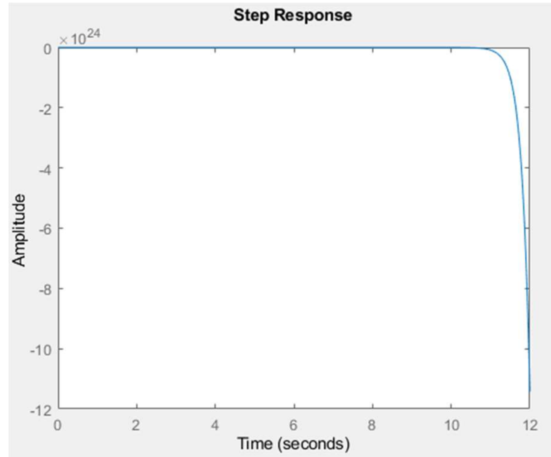
The step response is unbounded as the closed loop transfer function is unstable due to the presence of RHP poles.



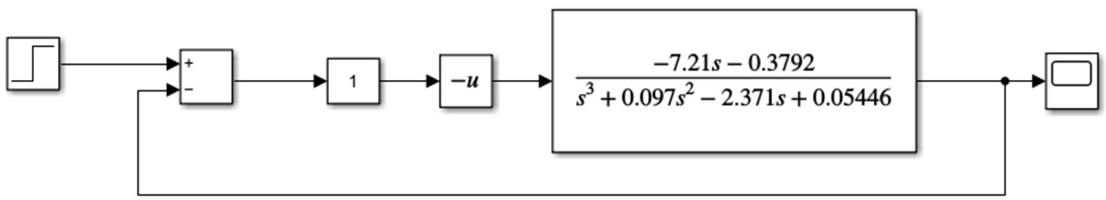
For k=2:



For k=3:

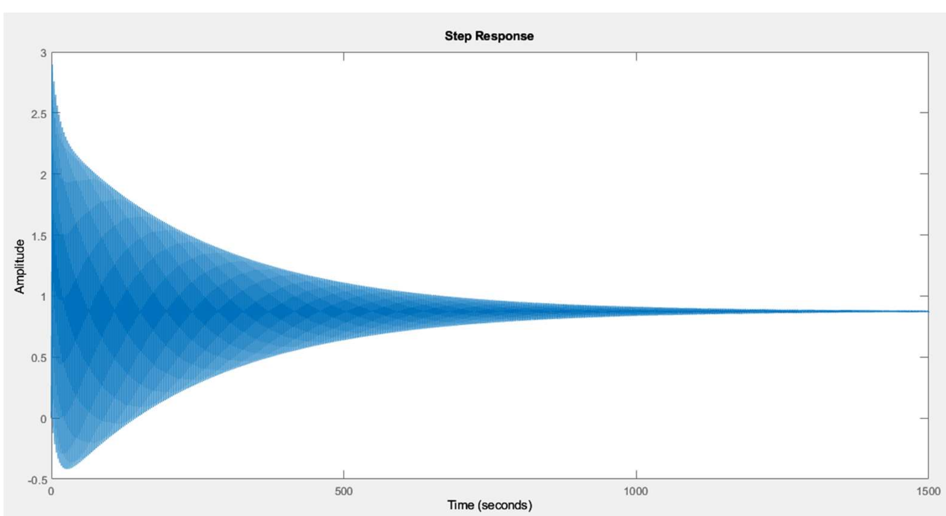


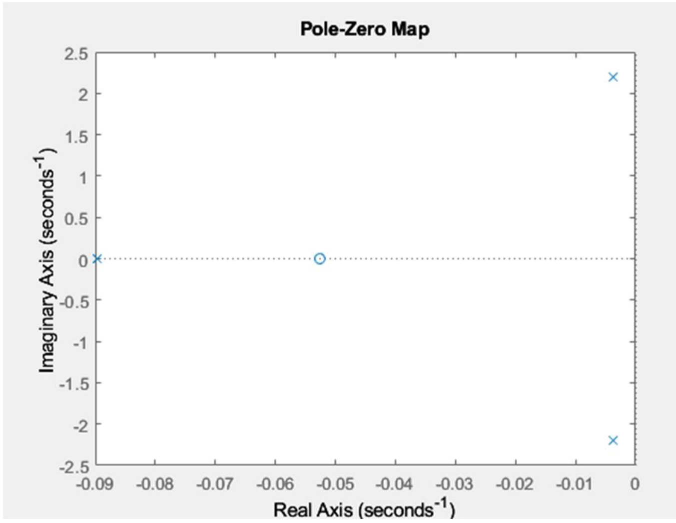
After further analysis, we conclude that the closed loop system is unstable for all positive values of the gain K. **Hence we place an inverter in cascade to the gain K to make the system closed loop stable.**



With the inverter in cascade with the gain K:

K=1:





From the pole zero map, we infer that the system is now closed loop stable with the following time domain parameters:

Rise Time: 0.3372s

Settling Time: $1.2068*10^3$s

Overshoot: 231.2%

Undershoot: 47.9%

Steady state error: -1.8967

*The range of K for which the system is closed loop stable, (**with the inverter in cascade**):*

The characteristic equation is given by, $1 + (-1)*K*G(s) = 0$

Which can be written as, $s^3 + 0.097s^2 + (-2.371+7.21K)s + (0.054 + 0.38K) = 0$

Using Routh-Hurwitz Criterion,

To find range of K for stability :

$$s^3 + 0.097s^2 + (-2.371 + 7.21K)s + (0.054 + 0.38K) = 0$$

s^3	1	$-2.371 + 7.21K$
s^2	0.097	$0.054 + 0.38K$
s^1	b_1	
s^0	$0.054 + 0.38K$	

$$b_1 = \frac{-1}{0.097}\left(0.054 + 0.38K - 0.097(-2.371 + 7.21K)\right)$$

$$= -10.31(0.284 - 0.319K)$$

$$= 3.29K - 2.93$$

For marginal stability, $b_1 = 0$

$3.29K - 2.93 = 0$

$K = 0.889$

b_1 is positive for values of $K > 0.889$.

Hence the system is closed loop stable for $\underline{K > 0.889}$

Auxillary equation →

$0.097S^2 + 0.054 + 0.38(0.889) = 0$

$0.097S^2 + 0.3918 = 0$

$\therefore S = \pm j2$

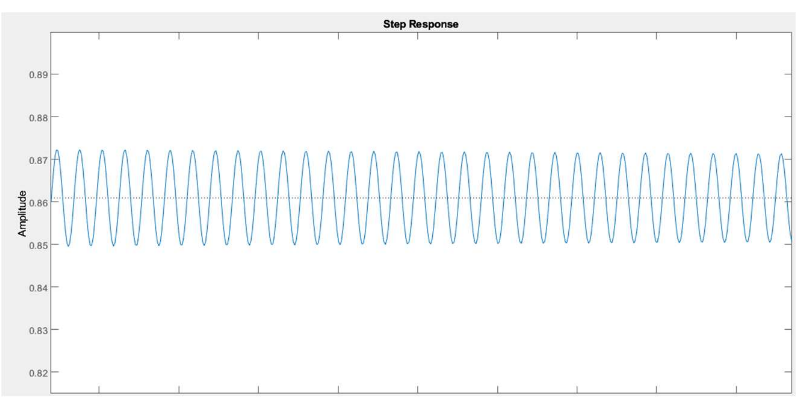
$\therefore \omega_M = 2$ rad/s

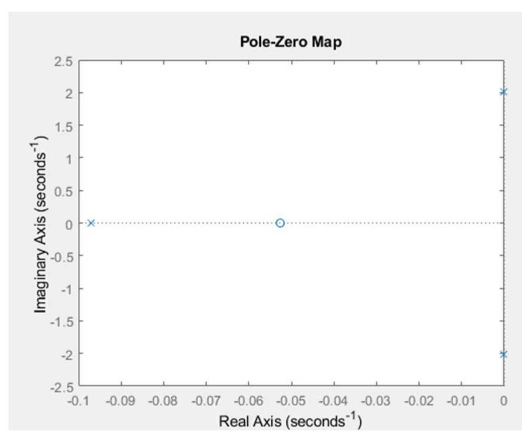
Range of K obtained after solving is K > 0.889.

The closed loop system is marginally stable at K=0.889.

The frequency of oscillations obtained is 2 rad/s.

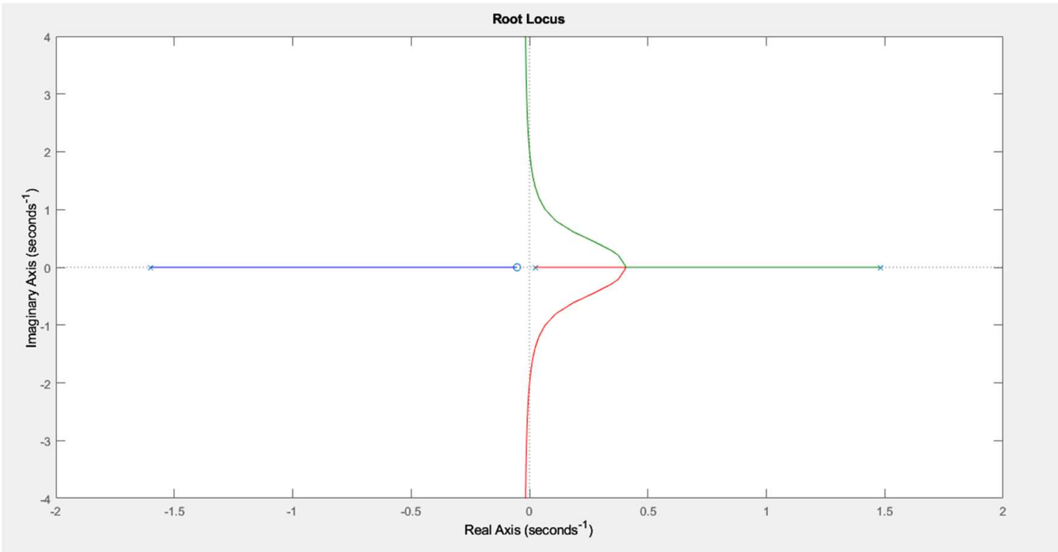
At K=0.889:



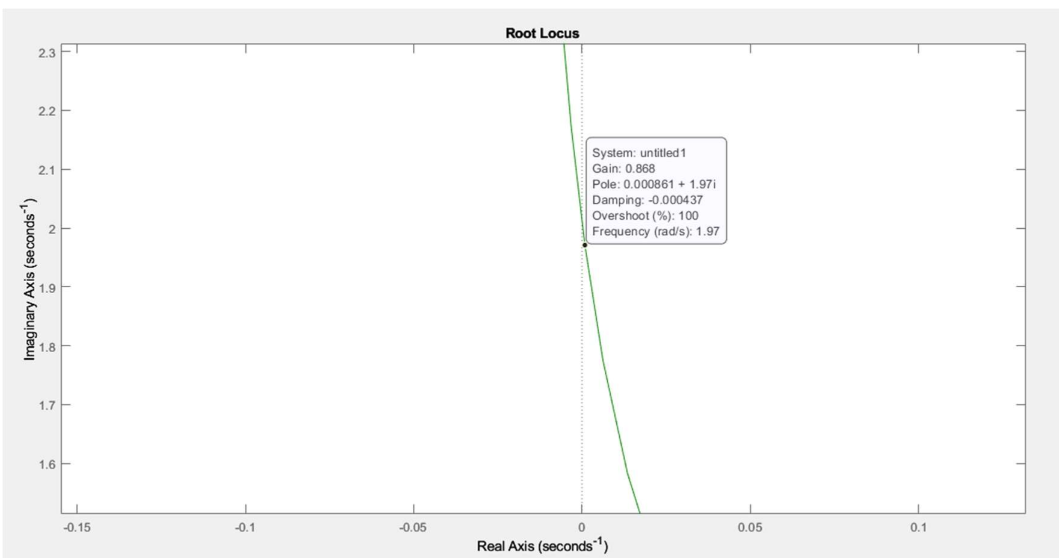


Conjugate poles are present on the jw axis, indicating marginal stability. Thus the value of K at which the output starts oscillating is K=0.889.

ROOT LOCUS



The plot obtained above is the root locus of the characteristic equation of the closed loop system. *(1+ (-1) *G(s) = 0)* **(gain K isn't considered)**



The gain at the point of intersection of the root locus plot with the jw axis is K=0.868.

From the plot, system is stable for K>0.868, which is close to 0.889. This verifies the range of K we have obtained from the Routh-Hurwitz criterion.

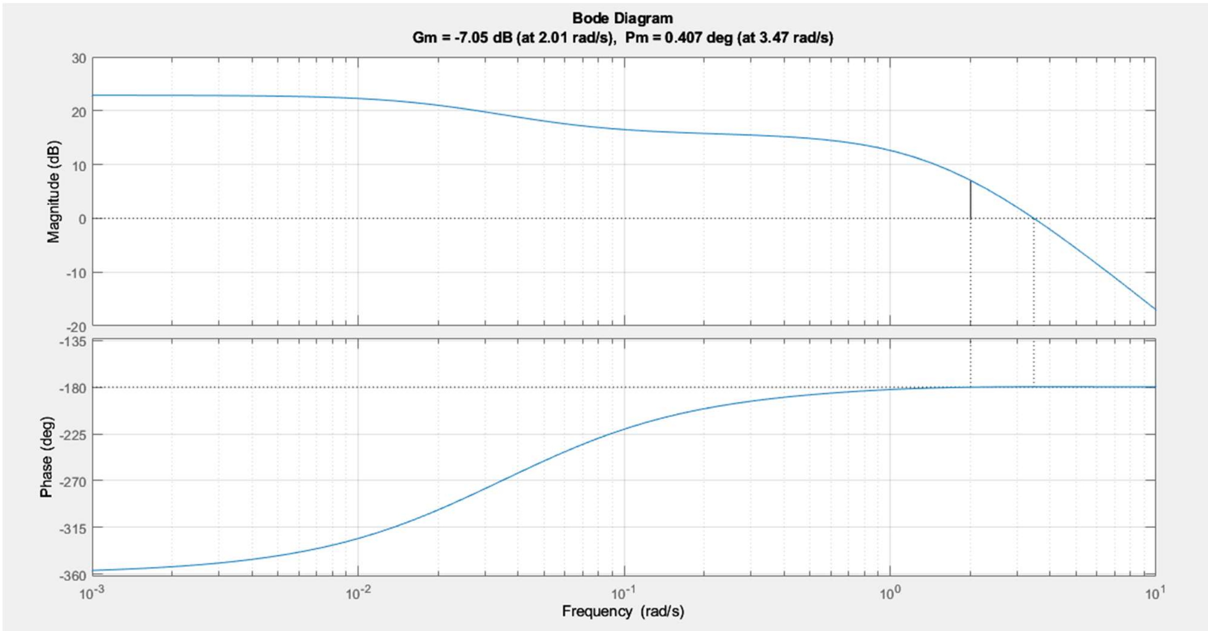
The frequency of oscillations obtained from the root locus is 1.97 rad/s, which is close to the theoretical value (2 rad/s).

DIFFERENCES IN SYSTEM BEHAVIOUR FROM OPEN LOOP TO CLOSED LOOP

The open loop system is unstable. On addition of a unity feedback path, and an inverter cascaded to the gain block K, the system becomes stable for K>0.889. The unit step response of the open loop system is unbounded in nature. While for the closed loop system, it oscillates and settles down to a steady state value (it is bounded). There is a significant reduction in the steady state error when unity feedback is used. By adding negative feedback to an open-loop system, the overall system becomes more stable, accurate and responsive to changes in the input.

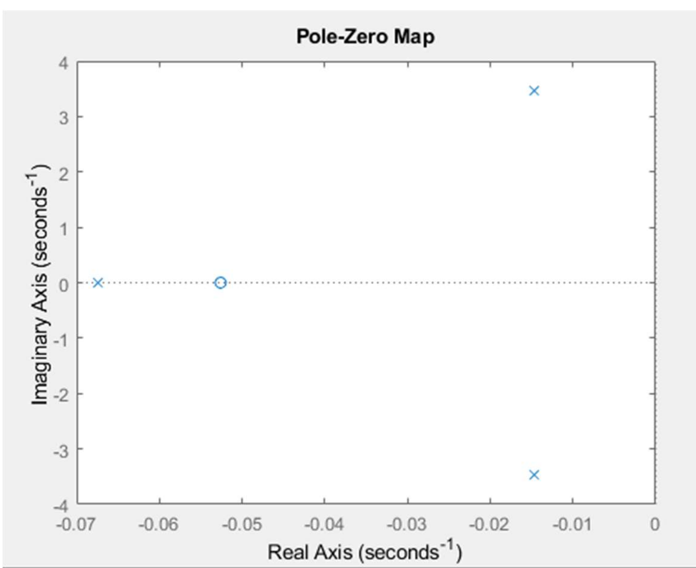
BODE PLOT

The value of gain K is chosen to be 2. (**Note that an inverter is placed in cascade to the gain**).

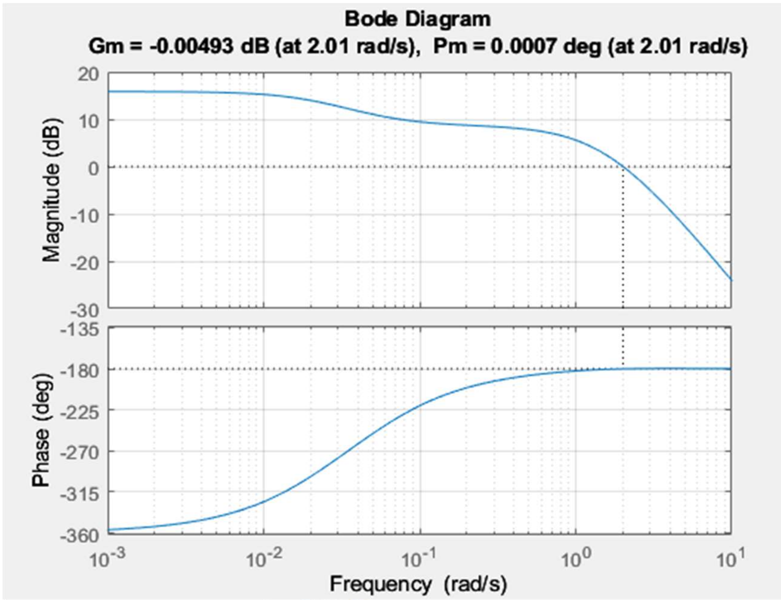


Gain margin = -7.05 dB, Phase margin = 0.407 degrees.

The phase margin is positive. Hence we can infer the system is closed loop stable. The pole zero map of closed loop system supports the inference:



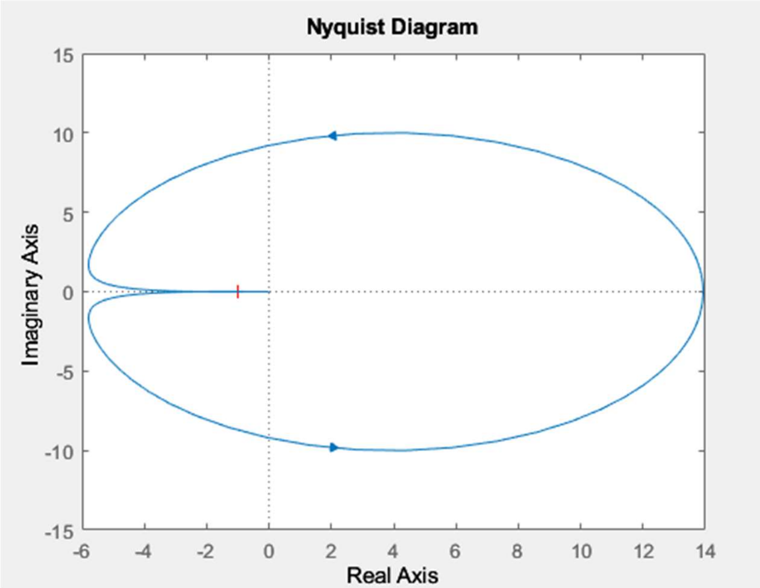
For K= 0.889,

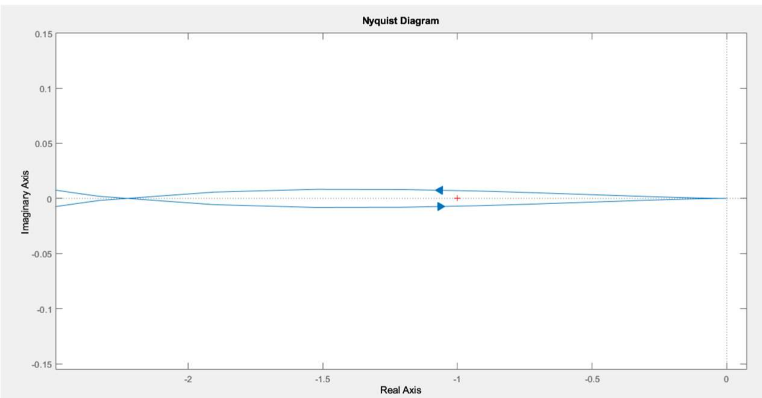


As gain margin is almost equal to the phase margin, system is marginally stable. This verifies the range of K obtained from the Root locus.

NYQUIST PLOT

The value of gain K is chosen to be 2 (with inverter in cascade).



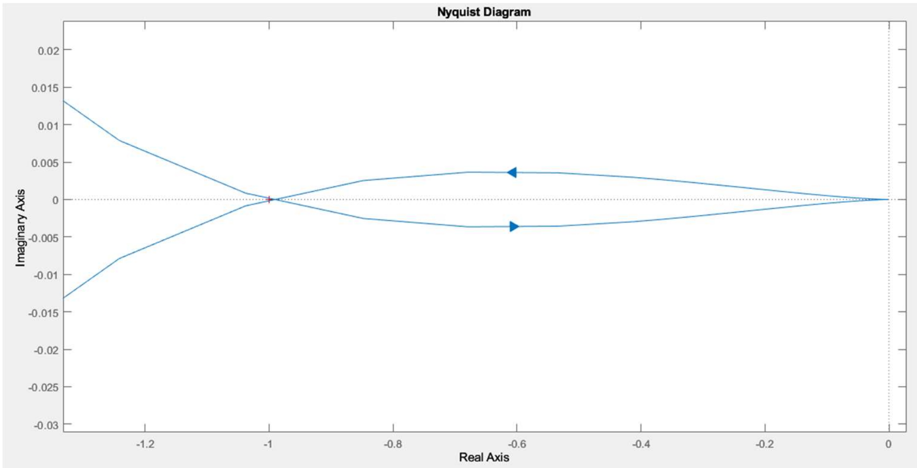


It is observed that the Nyquist plot encircles the -1 + j0 point twice in the counter clockwise direction. The number of right hand poles of the open loop transfer function is 2.

Thus N = -P = -2.

Z = N + P = 0. Thus we can infer that the system is closed loop stable.

Let us choose K=0.889, i.e, the point at which the system is marginally stable.



The plot almost touches the -1+j0 point. If we decrease K even further, we get an unstable system.

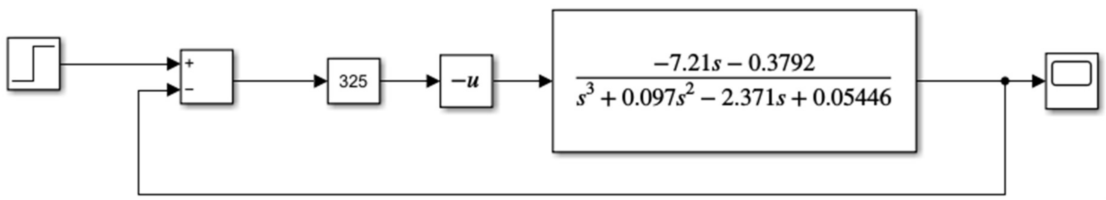
DESIGN OF CONTROLLERS

The variation in parameters is given by:

Gain/Parameters	***Overshoot***	***Settling Time***	***Steady State Error***
Increase in k_P	*Increases*	*Minimal Impact*	*Decreases*
Increase in k_I	*Increases*	*Increases*	0
Increase in k_D	*Decreases*	*Decreases*	*No Impact*

This table shall come in handy while fine tuning the controller.

PROPORTIONAL CONTROLLER



Theoretical Calculations:

$(s^2 + 2\omega_n \zeta s + \omega_n^2)(s+p)$

$p = 20$:

$(s^2 + 4s + 6.25)(s+20)$

$s^3 + 24s^2 + 86.25s + 125 = 0$

Comparing with,

$s^3 + 0.097s^2 + (7.21k - 2.371)s + (0.054 + 0.38k) = 0$

we get $k = 12.29$ or $k = 329$

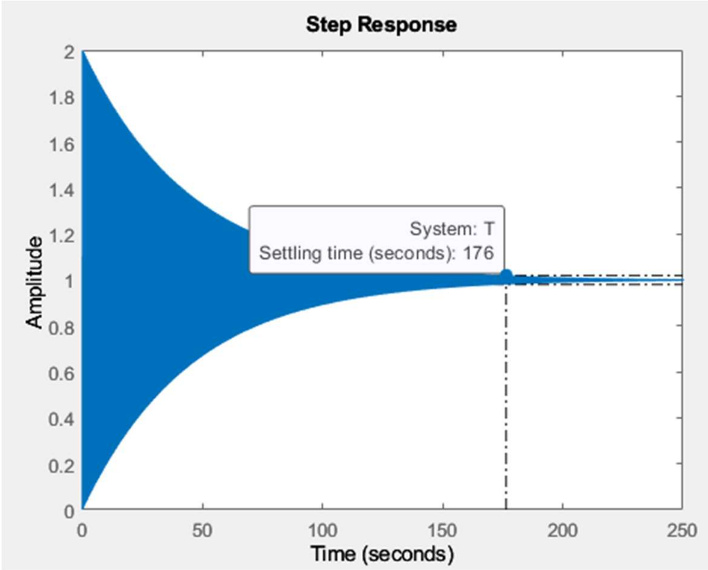
By comparing the actual characteristic equation with the desired characteristic equation, we get the value of the gain Kp as 325 after fine tuning the controller.

Kp = 325

The closed loop transfer function obtained is:

```
T =

          2343 s + 123.3
  --------------------------------
  s^3 + 0.097 s^2 + 2341 s + 123.3
```



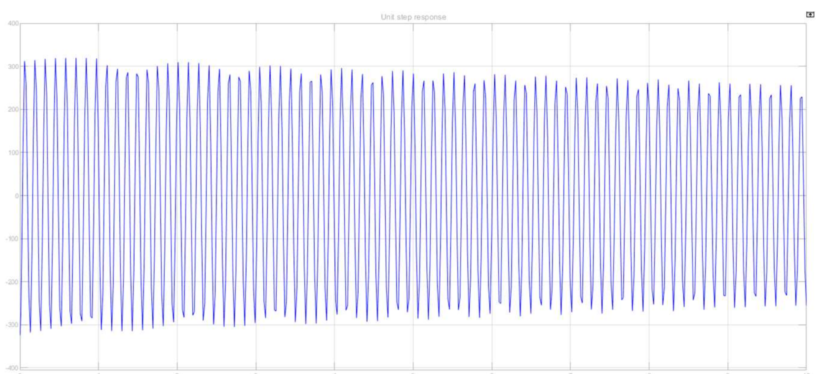
Settling time = 176 seconds

Peak Overshoot: 99.86%

From overshoot damping coefficient is obtained as 0.0032.

The controller makes the system closed loop stable but the obtained values are very far off the design specifications.

Controller effort:



PROPORTIONAL + DERIVATIVE CONTROLLER

Theoretical Calculations:

$$s^3 + 24s^2 + 86.25s + 125 = 0$$

comparing with,

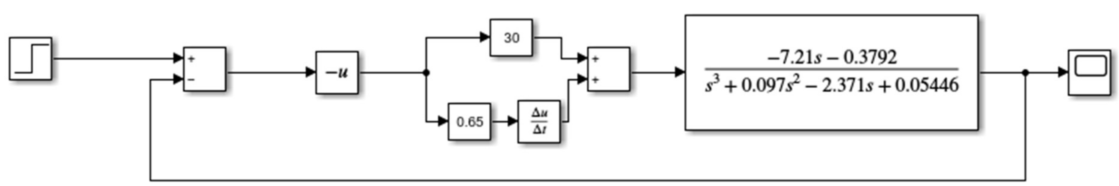
$$s^3 + (0.097 + 7.21K_d)s^2 + (7.21K_p + 0.38K_d - 2.371)s + (0.054 + 0.38K_p) = 0$$

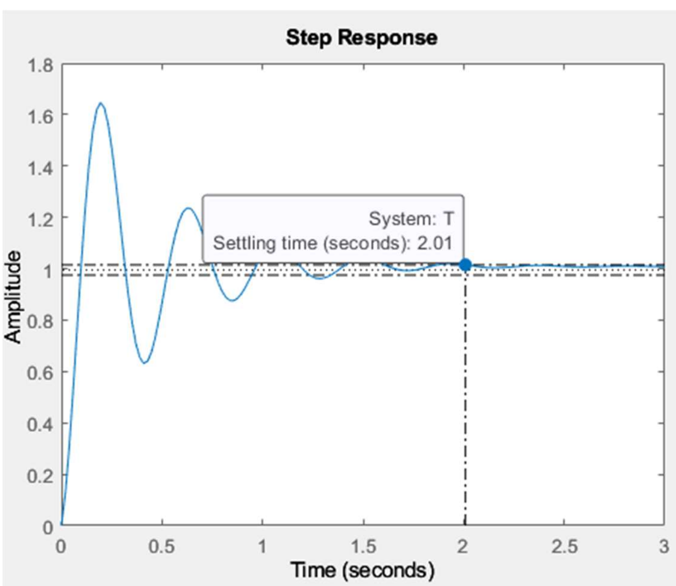
we get $K_p = 12.11$, $K_d = 3.33$ → Correct

$K_p = 329$, $K_d = -6009$ → Wrong

From the above theoretical values, we choose Kp= 12.11, Kd=3.33. A stable system is obtained but none of the desired specifications are met. Hence we tune the controller and consider two cases:

Case 1: We take Kp=30, Kd=0.65.





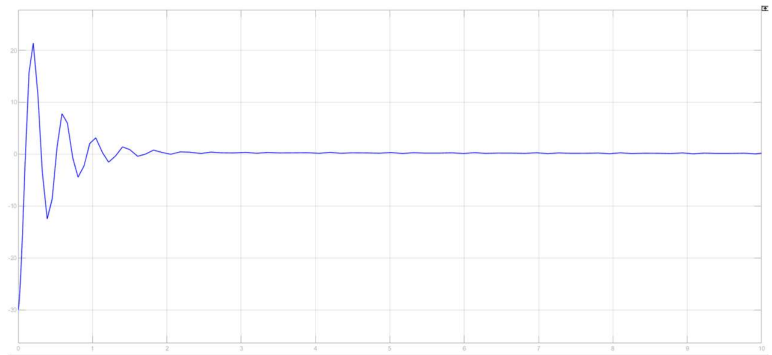
Settling time = 2.01 seconds

Peak Overshoot: 64.47%

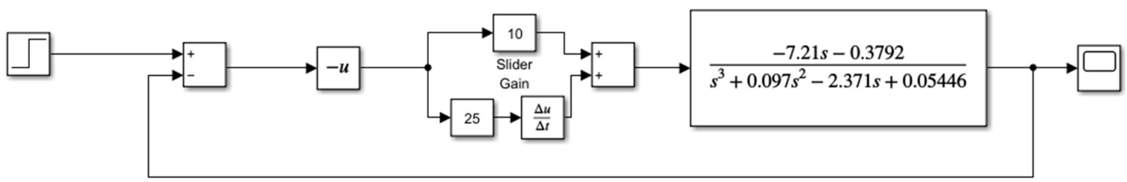
From overshoot damping coefficient is obtained as 0.14.

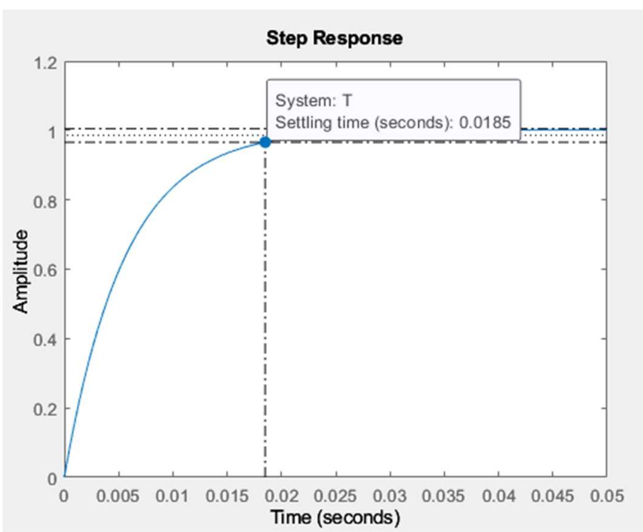
Desired settling time is obtained, but the damping coefficient is very low.

Controller effort:



Case 2: We take Kp=10, Kd=25





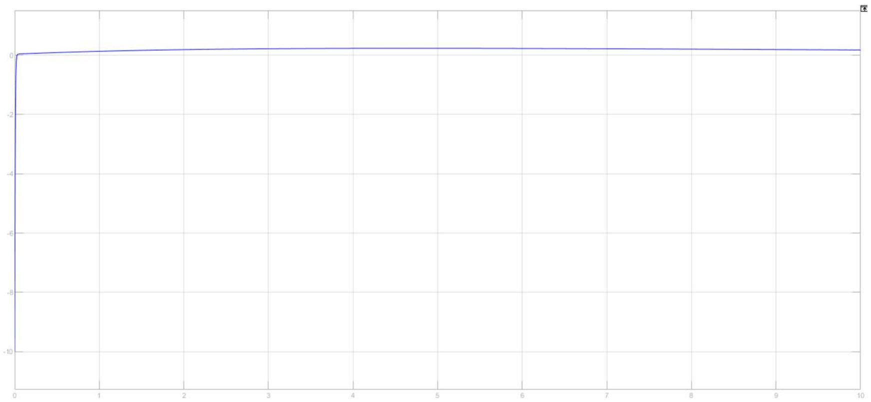
Settling time = 0.0185 seconds

Peak Overshoot:1.82%

From overshoot damping coefficient is obtained as 0.79.

Desired damping coefficient is obtained, but the settling time is very low.

Controller effort:



In the first case, we took the values of Kp and Kd such that settling time was accurate. In the second case the damping coefficient obtained was accurate. There is a trade-off between settling time and damping coefficient. Unfortunately both the desired specifications can't be obtained simultaneously using a PD controller.

Note: The use of a differentiator in the block diagram is for representative purposes only. For the actual design and analysis, a PID block has been used, by setting the value of Ki to 0.

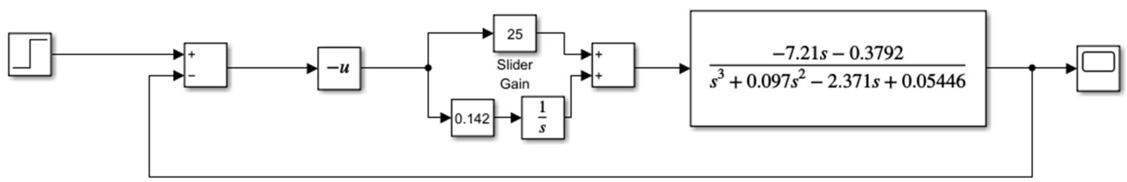
PROPORTIONAL + INTEGRAL CONTROLLER

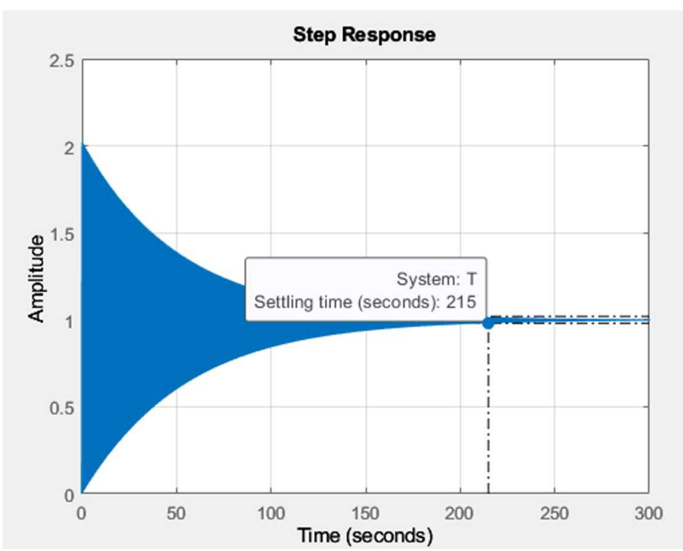
Theoretical Calculations:

$$(s^3 + 24s^2 + 86.25s + 125)(s+b) = 0$$
$$b = 6$$
$$s^4 + 30s^3 + 230.25s^2 + 642.5s + 750 = 0$$
Comparing with,
$$s^4 + 0.097s^3 + (7.21K_p - 2.371)s^2 + (0.38K_p + 0.054 + 7.21K_i)s + 0.38K_i = 0$$
$$K_p = 32.26, \quad K_i = 87.33$$
But system becomes unstable for these values.

If we use the theoretical values obtained, we get an unstable system due to the high value of Ki. Hence we tune the controller to obtain the following values:

Ki= 0.142, Kp= 25





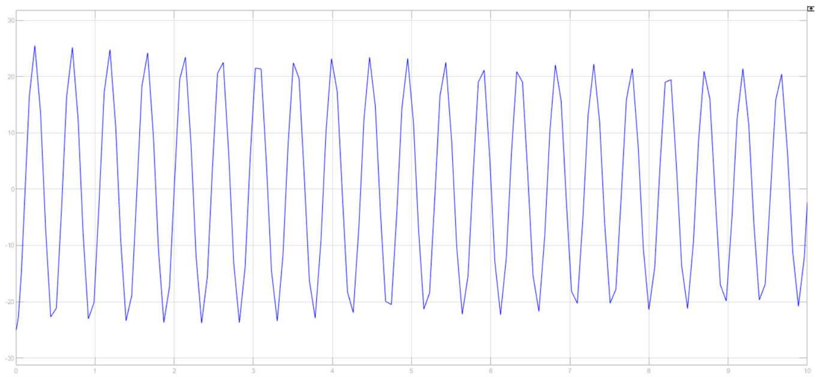
Settling time = 215 seconds

Peak Overshoot: 99%

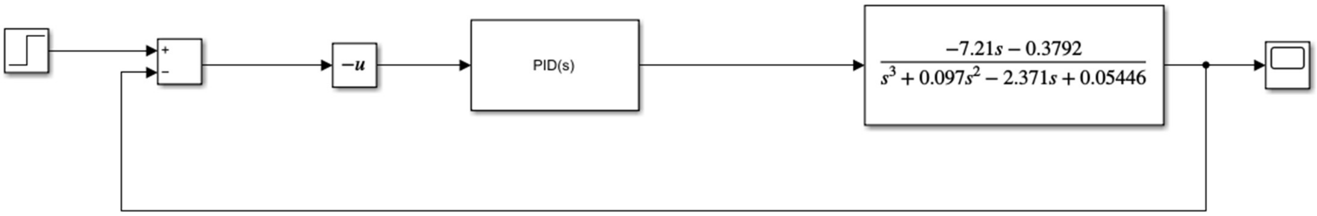
From overshoot damping coefficient is obtained as 0.0032.

System is closed loop stable but the values are extremely far off the design specifications. If we try to increase Ki, the system becomes unstable. Hence this type of controller is ineffective for our transfer function.

Controller effort:



PID CONTROLLER



Theoretical Calculations:

$(s^3 + 24s^2 + 86.25s + 125)(s+b) = 0$

$b = 12$

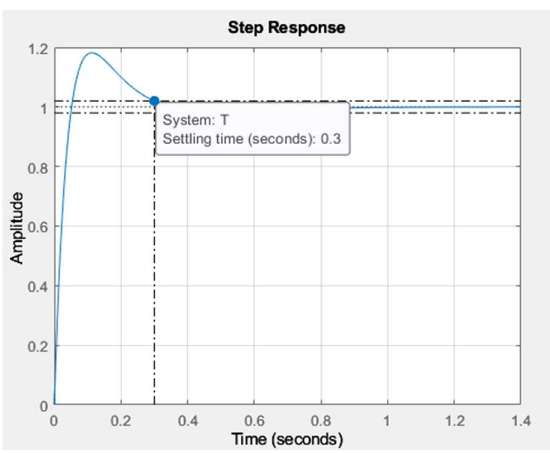
$s^4 + 36s^3 + 374.2s^2 + 1160s + 1500 = 0$

Comparing with,

$s^4 + (0.097 + 7.21K_d)s^3 + (-2.371 + 7.21K_p + 0.38K_d)s^2 + (0.38K_p + 0.054 + 7.21K_i)s + 0.38K_i = 0$

we get $K_p = 51$, $K_d = 5$, $K_i = 158$

The step response obtained for the theoretical values:



We get the settling time as 0.3 seconds. To obtain the desired settling time of 2 seconds, we manually tune the controller to get the below mentioned values:

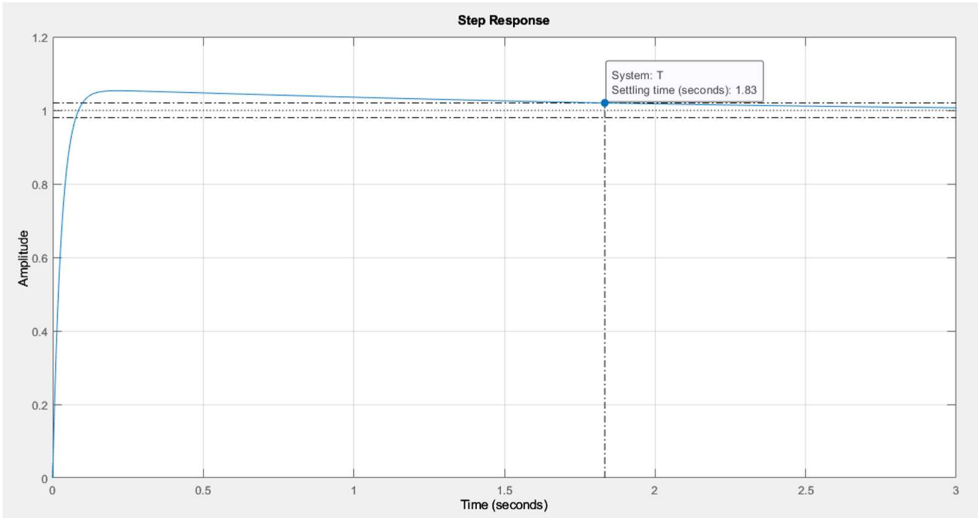
Details of the Controller:

Kp= 10

Kd= 4.9

Ki= 4.5

Constant Ki is reduced considerably.



Settling time = 1.83 seconds

Peak Overshoot: 5.32%

From overshoot damping coefficient is obtained as 0.69.

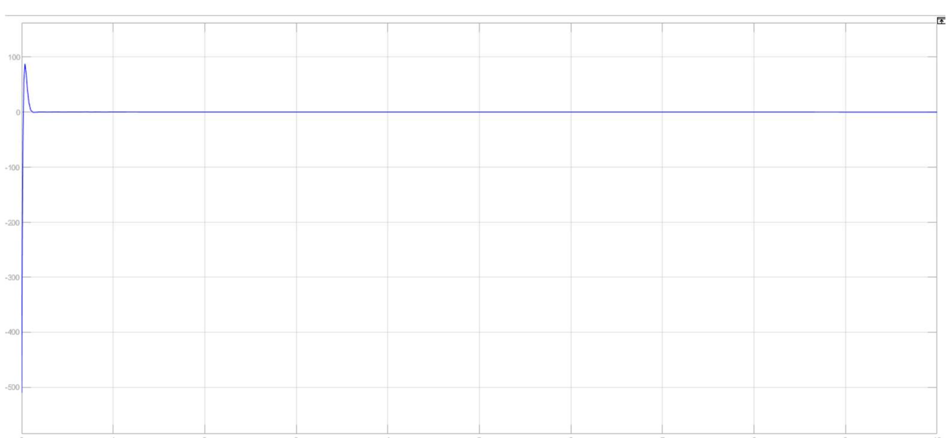
The values obtained are very close to the desired design specifications.

The closed loop transfer function obtained is:

```
T =

      36.05 s^3 + 74 s^2 + 34.8 s + 1.631
  ---------------------------------------------
  s^4 + 36.15 s^3 + 71.63 s^2 + 34.85 s + 1.631
```

Controller effort:



ROBUSTNESS CHECK

Case 1:

The co-efficients in the plant transfer function are changed by + 30 %.

New Open loop (or) Plant transfer function is given by,

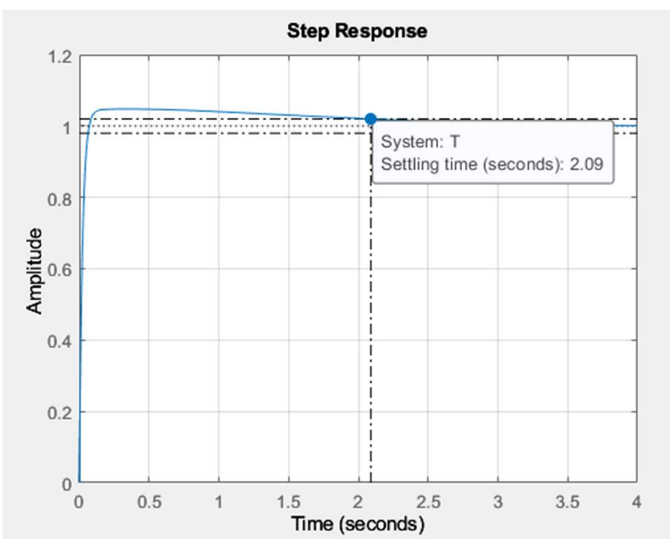
```
open_sys =

          -9.373 s - 0.4927
  ---------------------------------
  s^3 + 0.126 s^2 - 4.007 s + 0.1201
```

New Closed loop transfer function is given by,

```
T =

     45.93 s^3 + 96.14 s^2 + 47.11 s + 2.217
  ------------------------------------------
  s^4 + 46.05 s^3 + 92.14 s^2 + 47.23 s + 2.217
```



Settling time = 2.09 seconds

Peak Overshoot: 4.76%

From overshoot damping coefficient is obtained as 0.71.

Case 2:

The co-efficients in the plant transfer function are changed by - 30 %.

New Open loop (or) Plant transfer function is given by,

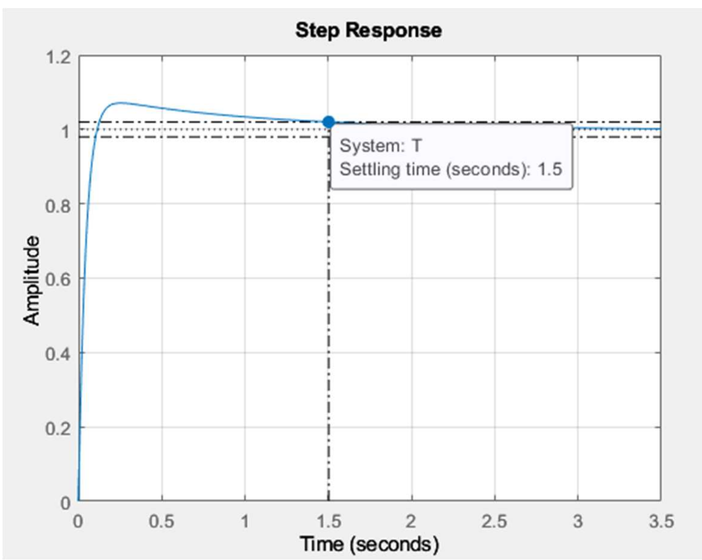
```
open_sys =

          -5.047 s - 0.2443
  ------------------------------------
  s^3 + 0.0679 s^2 - 1.162 s + 0.01868
```

New Closed loop transfer function is given by,

```
T =

    24.73 s^3 + 51.67 s^2 + 25.15 s + 1.099
  ------------------------------------------
  s^4 + 24.8 s^3 + 50.51 s^2 + 25.17 s + 1.099
```



Settling time = 1.5 seconds

Peak Overshoot: 7.1%

From overshoot damping coefficient is obtained as 0.62.

The design specifications obtained do not stray very far off the required values.

Thus, we can conclude that the design specifications are met, while there is an uncertainty in the plant parameters.

MATLAB CODE

Analysis of the open loop transfer function:

```
clc; close; clear all;
num = [-7.21 -0.379246];
den1 = [ 1 1.6];
den2 = [ 1 -1.48];
den3 = [ 1 -0.023];
den= conv((conv(den1,den2)), den3);
open_sys=tf(num,den);
s=tf('s');
figure
step(open_sys)
figure
step(open_sys/s) %unit ramp response
figure
pzmap(open_sys)
info = stepinfo(open_sys)
```

Analysis of the unity feedback system with a cascaded gain block K:

```
clc; close; clear all;
num = [-7.21 -0.379246];
den1 = [ 1 1.6];
den2 = [ 1 -1.48];
den3 = [ 1 -0.023];
den= conv((conv(den1,den2)), den3);
open_sys= (-1)*tf(num,den); % multiplication by -1 is done as we have placed an
inverter in cascade to the gain block
k= 2;
h=1;
T=feedback(k*open_sys,h)
figure
step(T)
figure
pzmap(T)
info = stepinfo(T)
ss_error = 1-(info.SettlingMax)
```

Root Locus:

```
%root locus
clc; clear all; close all;
num = [-7.21 -0.379246];
den1 = [ 1 +1.6];
den2 = [ 1 -1.48];
den3 = [ 1 -0.023];
den= conv((conv(den1,den2)), den3);
open_sys=(-1)*tf(num,den); % multiplication by -1 is done as we have placed an
inverter in cascade
rlocus(open_sys)
```

Bode Plot and Nyquist Plot:

```
clc; close; clear all;
num = [-7.21 -0.379246];
den1 = [ 1 1.6];
den2 = [ 1 -1.48];
den3 = [ 1 -0.023];
den= conv((conv(den1,den2)), den3);
open_sys= (-1)*tf(num,den);
k= 2; % we also took k=0.889 (marginal stability value)
h=1;
T=feedback(k*open_sys,h)
figure, margin(k*open_sys)
figure, nyquist(k*open_sys)
```

Design of PID Controller:

```
%PID Controller
clc; clear all; close all;
num = [-7.21 -0.379246];
den1 = [ 1 +1.6];
den2 = [ 1 -1.48];
den3 = [ 1 -0.023];
den= conv((conv(den1,den2)), den3);
open_sys=(-1)*tf(num,den);
kp= 10;
kd= 4.9;
ki= 4.5;
gc=pid(kp, ki, kd);
% For Proportional controller, kd and ki are set to zero
% For PD controller, ki is set to zero
% For PI controller, kd is set to zero
h=1;
T=feedback(gc*open_sys,h)
figure
pzmap(T)
figure
step(T)
grid on
stepinfo(T)
```

Robustness Check:

```
%ROBUSTNESS TEST +30 percent
clc; clear all; close all;
num = [-9.373 -0.4927];
den1 = [ 1 +2.08];
den2 = [ 1 -1.924];
den3 = [ 1 -0.03];
den= conv((conv(den1,den2)), den3);
open_sys=(-1)*tf(num,den)
kp= 10;
kd= 4.9;
ki= 4.5;
gc=pid(kp, ki, kd);
h=1;
T=feedback(gc*open_sys,h)
```

```
figure
pzmap(T)
figure
step(T)
grid on
stepinfo(T)


%ROBUSTNESS TEST -30 percent
clc; clear all; close all;
num = [-5.047 -0.2443];
den1 = [ 1 +1.12];
den2 = [ 1 -1.036];
den3 = [ 1 -0.0161];
den= conv((conv(den1,den2)), den3);
open_sys=(-1)*tf(num,den)
kp= 10;
kd= 4.9;
ki= 4.5;
gc=pid(kp, ki, kd);
h=1;
T=feedback(gc*open_sys,h)
figure
pzmap(T)
figure
step(T)
grid on
stepinfo(T)
```

REFERNCES

[1] Cem Çatalbaş, Arif Gulten, "A Novel Approach for Optimization of Nozzle Angle and Thrust Vectoring Controller via a Sub-Mutation Genetic Algorithm", in *International journal of innovative computing, information & control: IJICIC*

[2] Richard C. Dorf and Robert H. Bishop, *Modern Control Systems,* 13th ed. Pearson

[3] *https://illinois.edu/research/index.html*

[4] Katsuhiko Ogata, *Modern Control Engineering,* 5th ed. Pearson