Class:	EE370L Control Systems – 1002		Semester:	Spring 2023
Points		Document author:	**Maxwell Stonham** **Abdul Hamid Viar**	
		Author's email:	stonham@unlv.nevada.edu hamida1@unlv.nevada.edu	
		Document topic:	**Final Project**	
Instructor's comments:				

1. Introduction / Theory of Operation

Root Locus:

In control theory and stability theory, **root locus analysis** is a graphical method for examining how the roots of a system change with variation of a certain system parameter, commonly a gain within a feedback system. This is a technique used as a stability criterion in the field of classical control theory developed by Walter R. Evans which can determine stability of the system. The root locus plots the poles of the closed loop transfer function in the complex *s*-plane as a function of a gain parameter.

In addition to determining the stability of the system, the root locus can be used to design the damping ratio (ζ) and natural frequency (ω_n) of a feedback system. Lines of constant damping ratio can be drawn radially from the origin and lines of constant natural frequency can be drawn as arccosine whose center points coincide with the origin. By selecting a point along the root locus that coincides with a desired damping ratio and natural frequency, a gain K can be calculated and implemented in the controller. More elaborate techniques of controller design using the root locus are available in most control textbooks: for instance, lag, lead, PI, PD and PID controllers can be designed approximately with this technique.

The definition of the damping ratio and natural frequency presumes that the overall feedback system is well approximated by a second order system; i.e. the system has a dominant pair of poles. This is often not the case, so it is good practice to simulate the final design to check if the project goals are satisfied.

PD Controller:

A PD controller is described by the transfer function:

$$K(s)=kp+kds=kd(s+kp-/kd)$$

A PD controller thus adds a single zero to the loop transfer function. The closed-loop characteristic polynomial is given as:
The phase contribution of the PD controller increases from 0∘
at low frequencies to 90∘

at high frequencies.
For practical reasons, a pole with a short time constant, Tf
, may be added to the PD controller. The pole helps limit the loop gain at high frequencies, which is desirable for disturbance rejection. The modified PD controller is described by the transfer function:

$$K(s)=kp+\frac{k_d s}{(Tfs+1)}$$

The modified PD controller is very similar to a first-order phase-lead controller; it is similarly employed to improve the transient response of the system.

PI Controller:
A PI controller is described by the transfer function:

$$K(s)= kp + kis = kp\ (s+ki/kp)/s$$

The PI controller thus adds a pole at the origin (an integrator) and a finite zero to the feedback loop. The presence of the integrator in the loop forces the error to a constant input to go to zero in steady-state; hence PI controller is commonly used in designing servomechanisms.
The controller zero is normally placed close to the origin in the complex s-plane. The presence of a pole–zero pair adds a closed-loop system pole with a large time constant. The zero location can be adjusted so that the contribution of the slow mode to the overall system response stays small.

The PI-PD Controller:
The PD and PI sections can be combined in a PI-PD controller as:

$$K(s) = (kp+\frac{k_i}{s})(1+kds) \quad \text{or} \quad K(s)=(kp+kds)\ (1+ki/s)$$

The PI-PD controller adds two zeros and an integrator pole to the loop transfer function. The zero from the PI part may be located close to the origin; the zero from the PD part is placed at a suitable location for desired transient response improvement.
The PI-PD controller is similar to a regular PID controller that is described by the transfer function:

$K(s)=k_p+k_ds+k_i/s \; = k_ds2+k_ps+k_is$

The PID controller imparts both transient and steady-state response improvements to the system. Further, it delivers stability as well as robustness to the closed-loop system.

Design Problem:

For the unity-feedback system with

$$G(s) = \frac{k}{(s+4)(s+6)(s+10)}$$

Do the following:

a. Design a controller that will yield no more than 25% overshoot and no more than a 2-second settling time for a step input and zero steady-state error for step and ramp inputs.

Uncompensated System Root Locus:

First, we draw the root locus of the uncompensated system.

Zeros: None

Poles: z= -4, z= -6, z= - 10

Centroid = {sum of poles-sum of zeros} ÷ {#of poles - #of zeros)

Centroid = [-4-6-10-] ÷3 = -20/3 = -6.66

Assuming K=1, expanding G(s) yields

$$G(s) = \frac{1}{(s^2+10s+24)(s+10)} = \frac{1}{s^3+20s^2+124s+240}$$

Using MATLAB, we see our dominant pole at $s=-2.708 \pm 6.138j$

We know $|G(s)|=1$, solving for the uncompensated K yields:

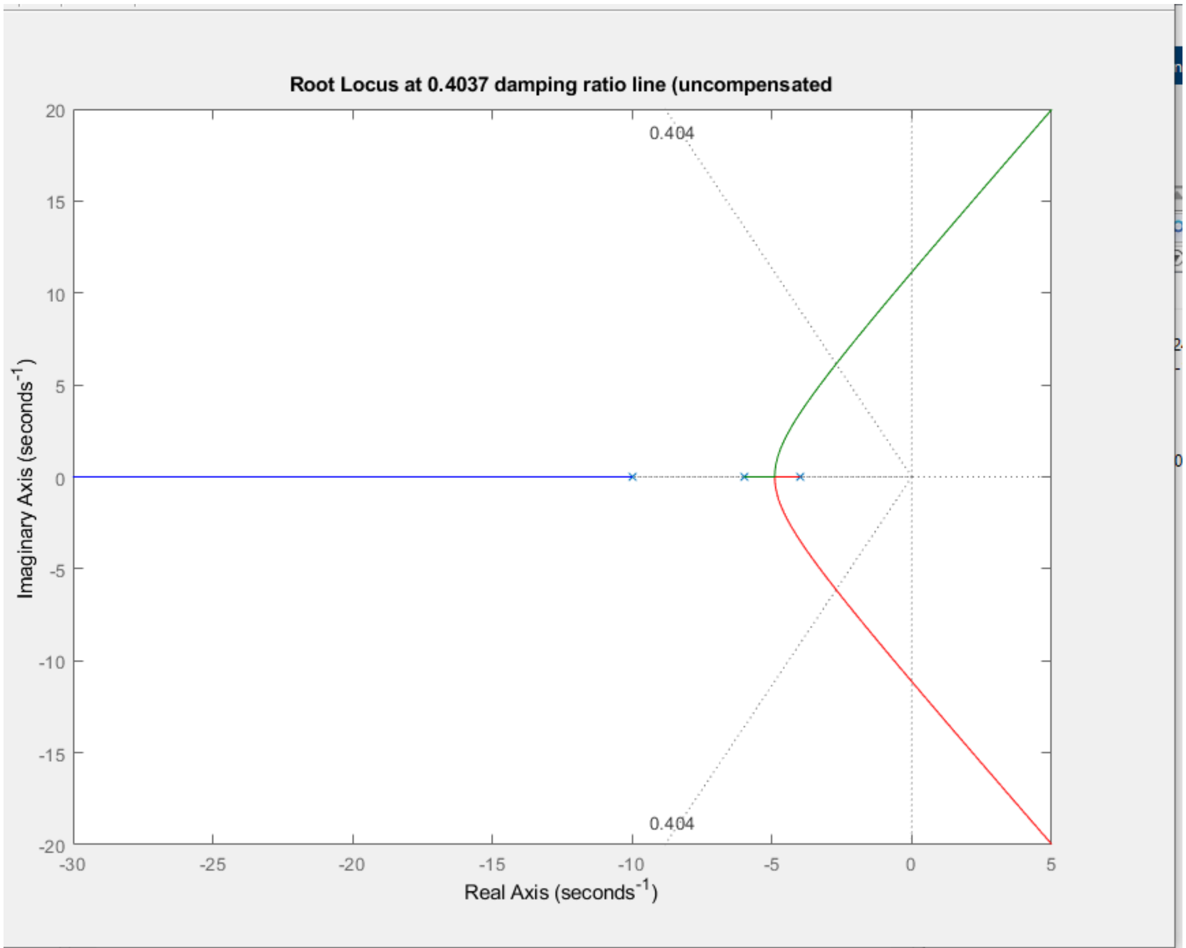
$$\left|\frac{K}{(s+4)(s+6)(s+10)}\right|_{s=-2.708\pm 6.138j} = 1$$

$$K = \sqrt{(1.292)^2+(6.138)^2}\cdot\sqrt{(3.292)^2+(6.138)^2}\cdot\sqrt{(7.292)^2+(6.138)^2}$$

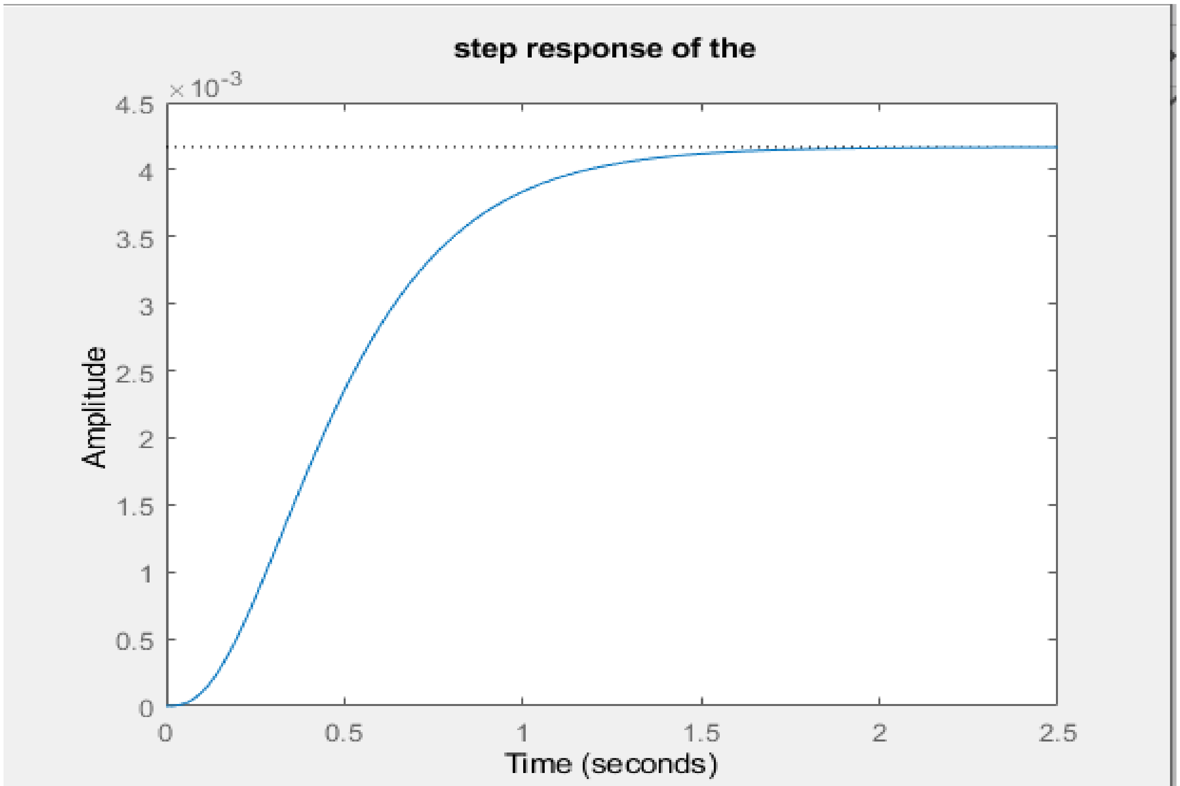
$$= (6.273)(6.965)(9.531)$$

$K = 416.42$ → uncompensated

- We can use MATLAB to draw the root locus:



Root Locus at 0.4037 damping ratio line (uncompensated
0.404
0.404
Imaginary Axis (seconds-1)
Real Axis (seconds-1)
20
15
10
5
0
-5
-10
-15
-20
-30
-25
-20
-15
-10
-5
0
5



Compensated System

Specifications:

%OS = 25%

Peak Time: Tp = 2 sec

At a damping ratio of 25%,

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} = \frac{-\ln(25/100)}{\sqrt{\pi^2 + \ln^2(25/100)}} \approx 0.4037$$

Damping Ratio = 0.4037

Steady State error for step input and ramp input = zero

Dominant pole can be obtained from the given specification:

For our compensated system, given a 2-second settling time:

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{\sigma} \rightarrow \sigma = \frac{4}{T_s} = \frac{4}{2} = 2 \rightarrow \text{real part of dominant pole}$$

We also know:

$$\omega_d = \sigma_d \tan(\cos^{-1}(\zeta)) = 2\tan(\cos^{-1}(0.4037))$$
$$= 4.533$$

So, our dominant pole is at $s = -2 \pm j4.533 \rightarrow$ compensated

In order to make the steady state error zero for **step input** and **ramp input**:

We know $G_{PD} = (s + z_c)$. Combining $G(s)$ with $G_{PD}(s)$ yields:

$$G_T(s) = G(s) \cdot G_{PD}(s) = \frac{K(s+z_c)}{(s+4)(s+6)(s+10)}$$

To satisfy a zero steady state error for step and ramp inputs, we need our G_{PD} to be $G_{PD} = \frac{s+z_c}{s}$, so now:

$$G_T(s) = \frac{K(s+z_c)}{s(s+4)(s+6)(s+10)}$$, where $\theta_p - \theta_z = 180°$ and $K=1$, and $s = -2 \pm j4.533$

$$[\angle s + \angle s+4 + \angle s+6 + \angle s+10] - \angle s+z_c = 180°$$

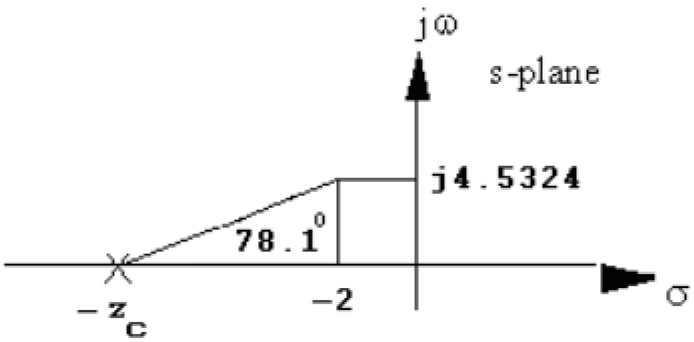
$$\angle -2 + j4.533 + \angle 2 + j4.533 + \angle 4 + j4.533 + \angle 8 + j4.533 - \angle -2 + j.4533 + z_c = 180°$$

$$\tan^{-1}\left(\frac{4.533}{-2}\right) + \tan^{-1}\left(\frac{4.533}{2}\right) + \tan^{-1}\left(\frac{4.533}{4}\right) + \tan^{-1}\left(\frac{4.533}{8}\right) - \tan^{-1}\left(\frac{4.533}{z_c - 2}\right) = 180°$$

$$-66.19° + 66.19° + 48.57° + 29.54° - \tan^{-1}\left(\frac{4.533}{z_c - 2}\right) = 180°$$

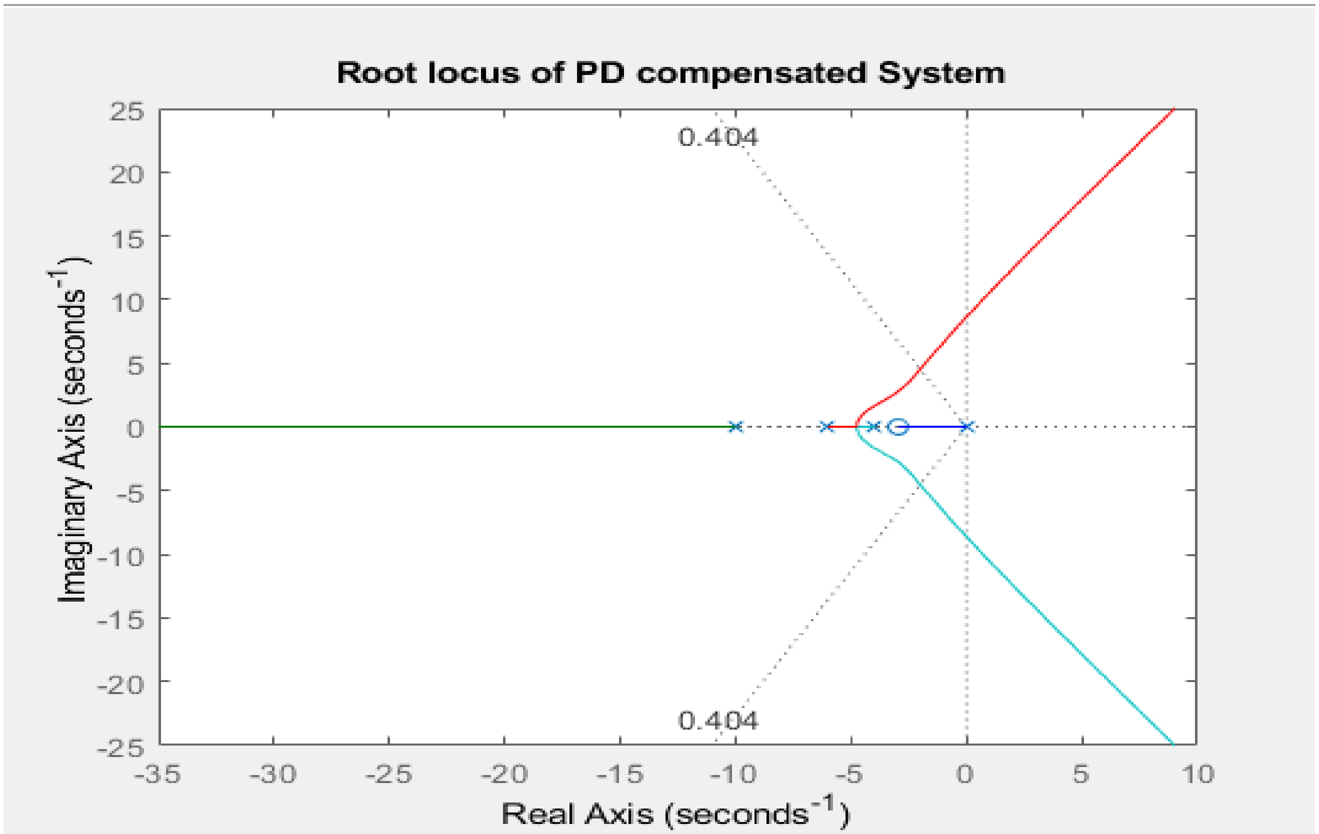
$$\tan^{-1}\left(\frac{4.533}{z_c - 2}\right) = -101.89° \rightarrow \frac{4.533}{z_c - 2} = 4.75 \rightarrow 0.9544 = z_c - 2$$

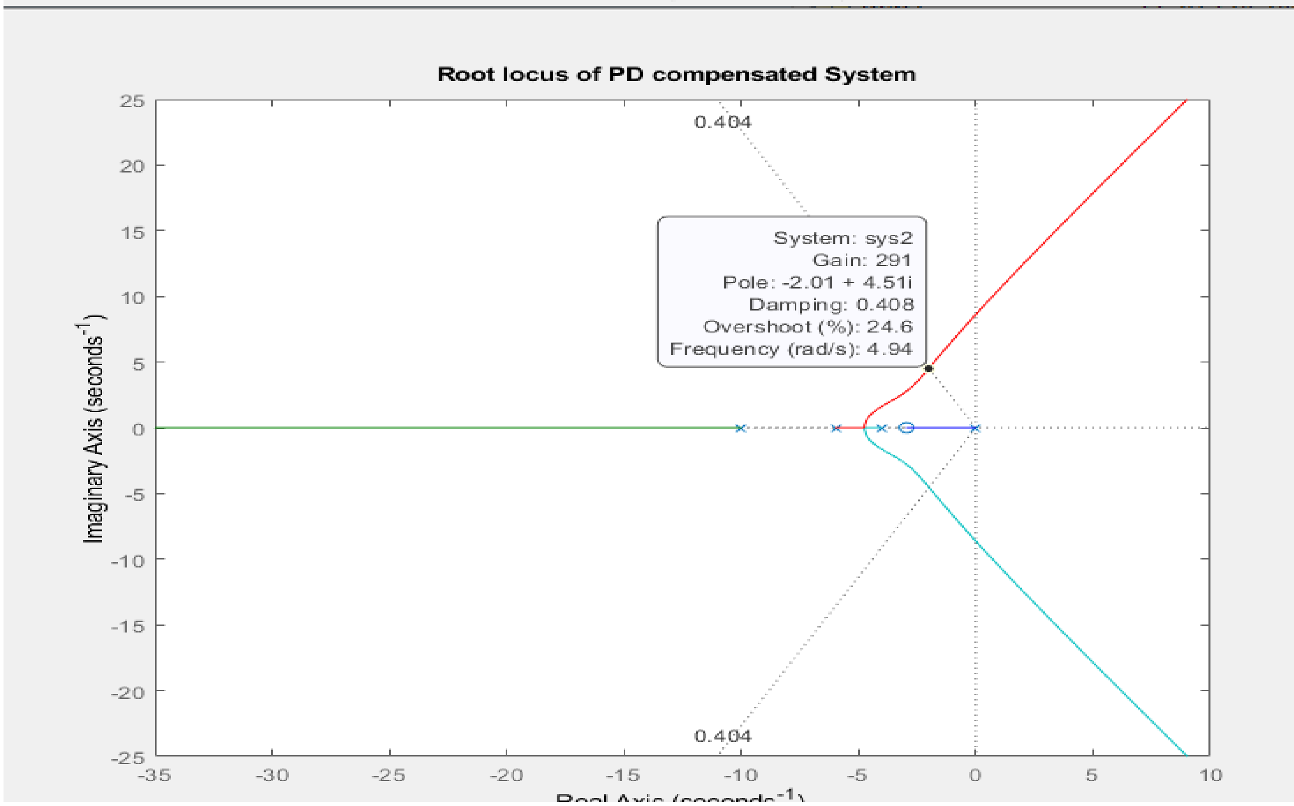
$$z_c = 2.9544$$



Matlab: we can use matlab to get the same results as from the calculation and hand plotting.

```
%% Root Locus of the PD Compensated System
num2 = [1 2.9544] ;
den2 = [1 20 124 240 0 ];
sys2 = tf(num2,den2)
figure(1);
rlocus(sys2)
title('Root locus of PD compensated System ')

% Damping ratio line
damping_ratio = 0.4037;
sgrid(damping_ratio, 0)
```


Root locus of PD compensated System
0.404
0.404
Imaginary Axis (seconds^-1)
Real Axis (seconds^-1)


Root locus of PD compensated System
0.404
0.404
System: sys2
Gain: 291
Pole: -2.01 + 4.51i
Damping: 0.408
Overshoot (%): 24.6
Frequency (rad/s): 4.94
Imaginary Axis (seconds^-1)

So now, $G_T(s) = \dfrac{K(s+2.9544)}{s(s+4)(s+6)(s+10)}$

We know $G_{PI} = \dfrac{(s+z)}{s}$ and choosing $z = 0.01$, so now:

$G_{PID}(s) = G_T(s) \cdot G_{PI}(s)$

$$G_{PID}(s) = \frac{K(s+2.9544)(s+0.01)}{s^2(s+4)(s+6)(s+10)}$$

Using the magnitude criteria and solving for K, (of $G_T(s)$)

$|G_T(s)| = 1$

$$K = \left| \frac{s(s+4)(s+6)(s+10)}{s+2.9544} \right|_{s=-2+j4.533}$$

$$= \frac{\sqrt{(2)^2+(4.533)^2} \cdot \sqrt{(2)^2+(4.533)^2} \cdot \sqrt{(4)^2+(4.533)^2} \cdot \sqrt{(8)^2+(4.533)^2}}{\sqrt{(0.9544)^2+(4.533)^2}}$$

$$= \frac{(4.955)(4.955)(6.046)(9.195)}{4.6324}$$

$$K = \frac{1364.919}{4.6324} = 294.65 \longrightarrow PD$$

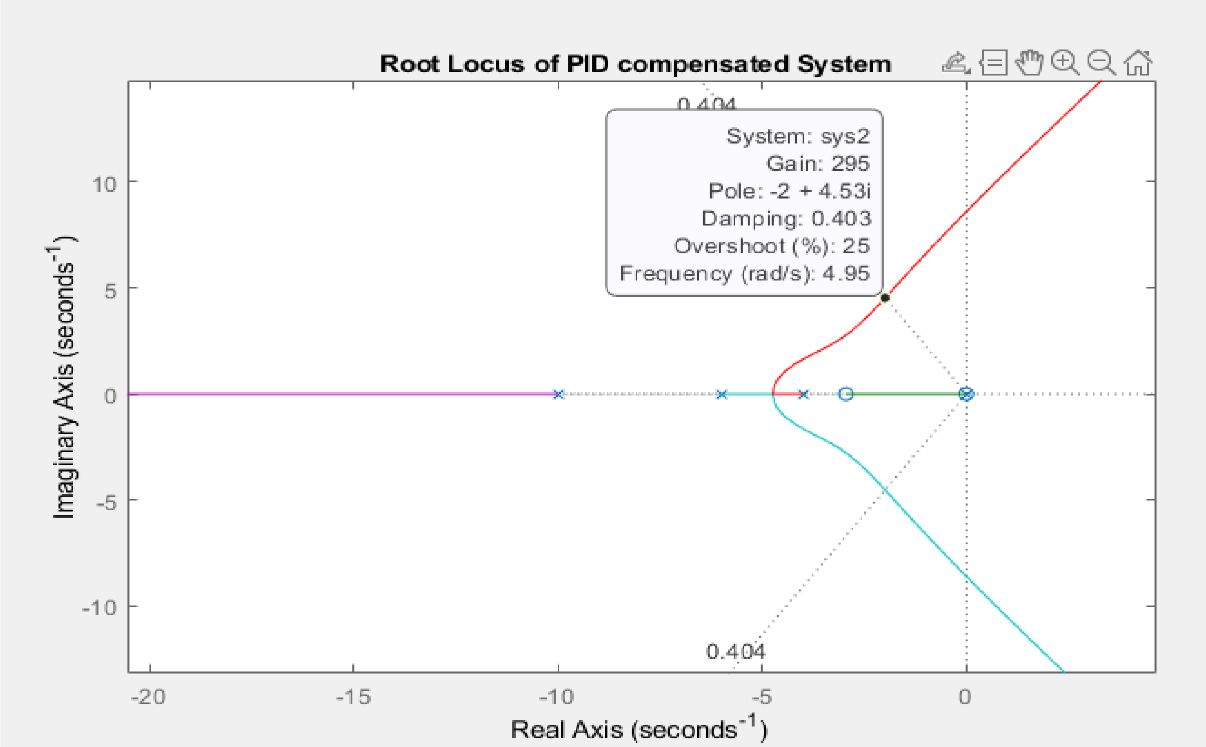
Similarly for $G_{PID}(s)$, our K becomes:

$$K = \left| \frac{s^2(s+4)(s+6)(s+10)}{(s+0.01)(s+2.9544)} \right|_{s=-2+j4.533}$$

$K = 294.9 \longrightarrow PID$

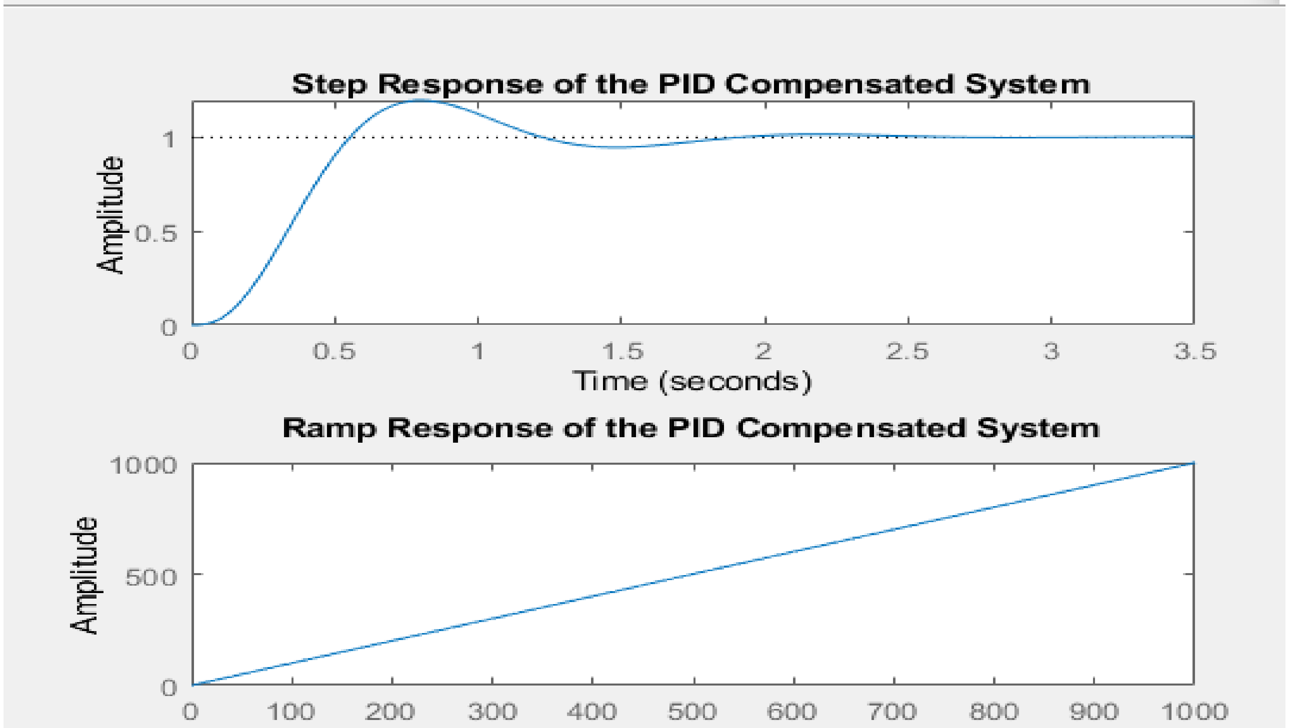
Matlab Results of the PID compensated System

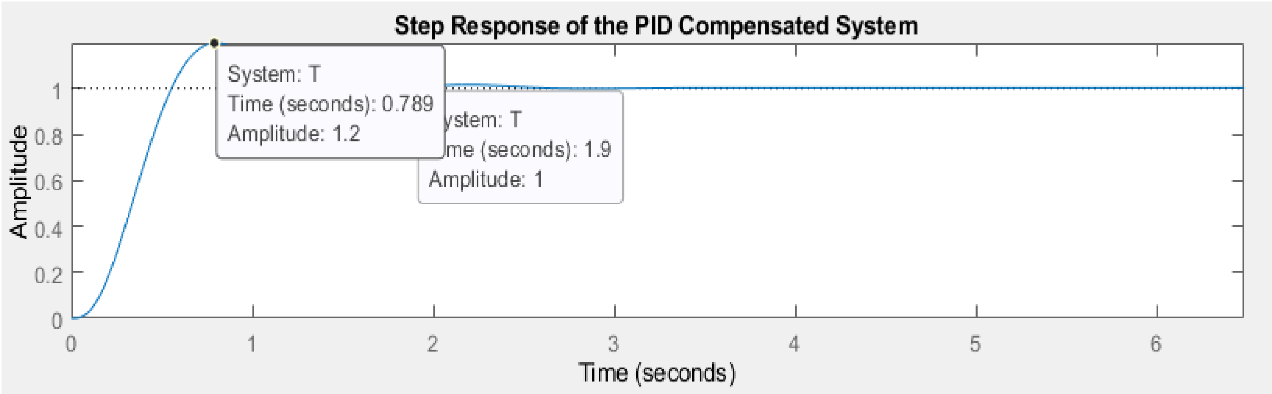
```
%% Root Locus of the PID Compensated System
num2 = [1 2.9644 0.0295] ;
den2 = [1 20 124 240 0 0 ];
sys2 = tf(num2,den2)
figure(3);
rlocus(sys2)
title('Root Locus of PID compensated System ')

% Damping ratio line
damping_ratio = 0.4037;
sgrid(damping_ratio, 0)
```



untitled9.m | titled3.m

```
%% Step Response of the PID Compensated System
numg=[-2.995 -0.01];
deng=[0 0 -4 -6 -10];
K=294.75;
G=zpk(numg,deng,K)
T=feedback(G,1);
figure(4)
subplot(2,1,1)
step(T)
title('Step Response of the PID Compensated System')
%% Ramp Response of the PID Compensated System
Ta=tf([1],[1 0]);

subplot(2,1,2)
step(T*Ta)
title('Ramp Response of the PID Compensated System')
```





So, our new compensated Transfer Function is:

$$G_T(s) = \frac{k(s+2.9544)(s+0.01)}{s2(s+4)(s+6)(s+10)}$$

Comparison of the Uncompensated and Compensated Systems

Parameters	Uncompenated	Compensated for integration (1/s)	Compensated for PD k(s+zc)/s	Compensated for PID
% OS	25%	25%	25.13%	25%
Damp ratio ζ	4.036	4.036	4.024	0.4037
Settling Time (sec)	1.47	2	2	2
Steady State error of step input	0.365	0	0	0
Steady State error of ramp input	∞	0.576	0.276	0