Birzeit University
Faculty of Engineering & Techonology
Department of Electrical & Computer Engineering
ENEE4302-Control Systems

“Active and Passive Compensators Design & Realization ”

Student :Mohamad Emad Bornat
#1130842

Instructor:Dr.Jamal Siam

January,2017.

Abstract

This work aims to investigate the design and implementation of active and passive compensators to correct for steady state error, settling time, and overshoot. Different controllers were designed and realized using passive and active circuits depending on the root locus of the open loop system with the help of Matlab.

Contents

Acronyms and Abbreviations

TF	Transfer Function
SSE	Steady State Error

List of Figures

Chapter 1

System Simulation

Plant G(s) with open–loop Transfer function was given to be :

$$\mathrm{G(s)} = \frac{\mathrm{s}+12}{(\mathrm{s}+10)(\mathrm{s}^2+2\mathrm{s}+8)}$$

Hence , The Transfer function for closed loop system H(s):

$$H(s) = \frac{KG(s)}{1+KG(S)H(S)} = \frac{K(S+12)}{S^3+18S^2+(18+K)s+28+10K}$$

Closed loop poles 1+KGH=0, so $GH = -\frac{1}{K}$.

By using code (A.0) The root locus plot of the system was found to be :

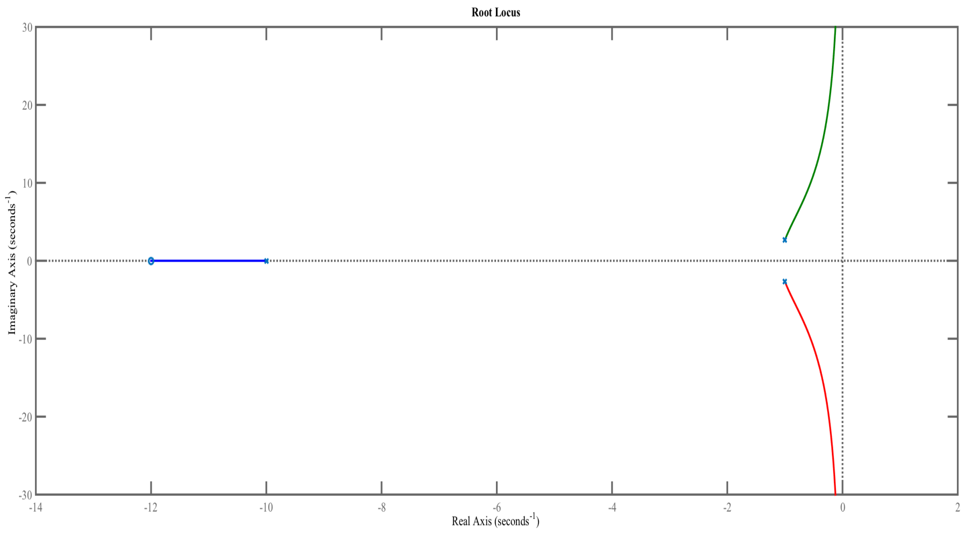

Fig. 1.1: Root Locus Plot of the Open Loop System.

1.1 Case 1: Improving Response Using PD Controller.

1.1.1 Controller Physical Realization and Design .

$$\mathrm{Gc(s)} = \mathrm{R2C}(\mathrm{s} + \frac{1}{\mathrm{R1C}})$$

$$\xi = \ln(0.18)/[(\pi^2 + \ln^2(0.18)]^{1/2} = 0.481$$

$$Ts = 4/\ \xi\omega_n \Rightarrow \omega_n = 4/(0.68*0.481) = 12.1 \text{ rad/sec .}$$

$$P_d = \xi\omega_n + j\omega_d$$

$$\omega_d = \omega_n(1-\xi^2)^{0.5} = 2.3094$$

$$P_d = -5.84 \pm j10.6$$

By using synthesis, we not that:

$$\mathrm{Gc(s)} = \mathrm{R2C}(\mathrm{s} + \frac{1}{\mathrm{R1C}})$$

$$\mathrm{K} = \mathrm{R2C1}\ \ ;\ \ \mathrm{Zc} = \frac{1}{\mathrm{R1C1}}$$

$$\mathrm{Gc(s)} = \mathrm{K(s + Zc)} = 10(\mathrm{s} + 12)$$

1.1.2 Controller Simulation.

using **PD Controller**, we can achieve the given specifications since the point is not located on the root locus.

At O.S =17.8%, the root locus gives the following:

Dominant second order poles: $\boldsymbol{-5.84 \mp j10.6}$
Gain (K) = 9.99

Second order approximation is valid since the third pole is far enough from the real part of the dominant pole, so the transient parameter formulas can safely be used in the design.By using code (A.1.1) The root locus plot of the system was found to be :

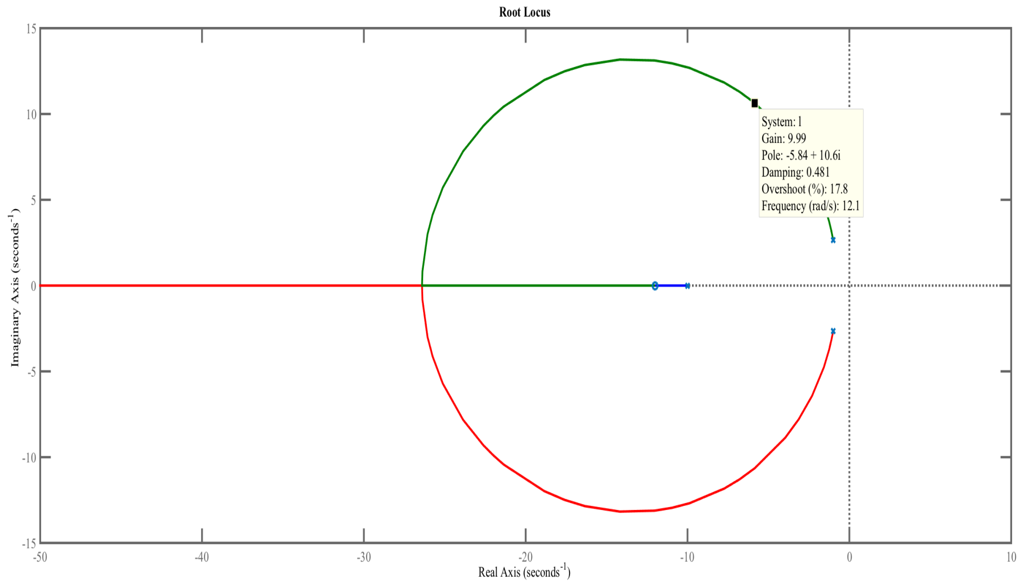

Fig. 1.2: Root Locus plot with PD controller K=10.

Now, the step response of the compensated system and the uncompensated system was checked :

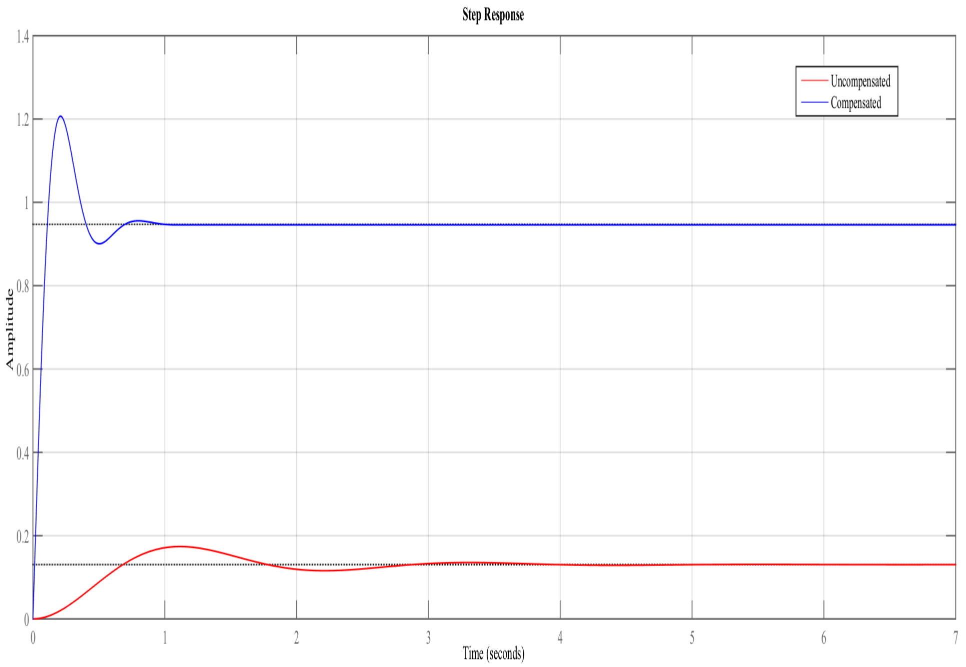

Fig. 1.3: step response of the compensated system and the uncompensated system with PD controller K=10.

By using code (A1.3) and by observing the figures ,then the following New Parameters were achieved :

Overshoot	17.8%
Settling Time	1/Re(pole)= 0.6826
Kp	18
Steady State Error ,ep	1\(1 + K_P)= 0.0526
Gain	9.99

1.2 Case 3: Improving Steady State Error to Zero using PID.

1.2.1 Controller Physical Realization .

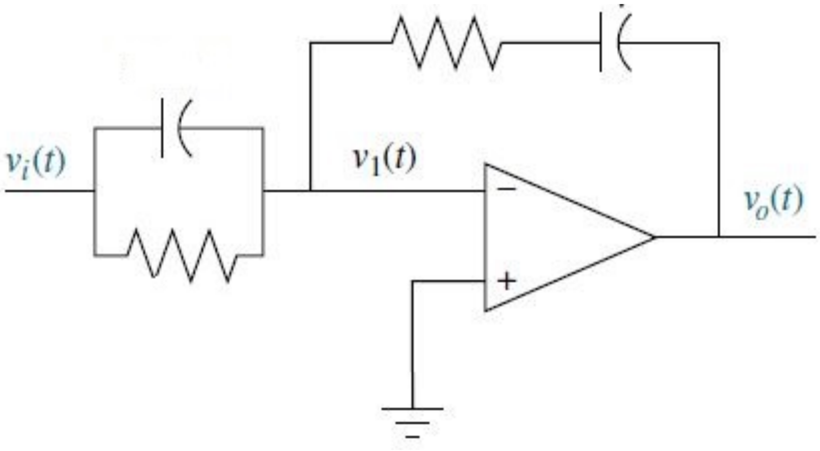

Fig. 1.4: PID Controller Implemetation.

$$\mathrm{Gc(s)} = \mathrm{K}\frac{(\mathrm{S} + \mathrm{Zlag}) + (\mathrm{S} + \mathrm{Zlead})}{\mathrm{R1C}}$$

- Lag zero at _zlag and pole at origin improve steady-state error.
- Lead zero at _zlead improves transient response.
- Lag zero at _zlag is close to, and to the left of the origin.
- Lead zero at _zlead is selected to put design point on root locus.

$$\mathrm{Gc(s)} = \mathrm{K}\frac{(\mathrm{S} + 0.002) + (\mathrm{S} + 15)}{\mathrm{S}}$$

1.2.2 Controller Simulation.

using **PID Controller**, we can achieve the given specifications since the point is not located on the root locus.

At O.S =19.2%, the root locus gives the following:

Dominant second order poles: $\mathbf{-3.5 \mp j6.66}$
Gain (K) = 4.95

By using code (A.2.1) The root locus plot of the system was found to be :

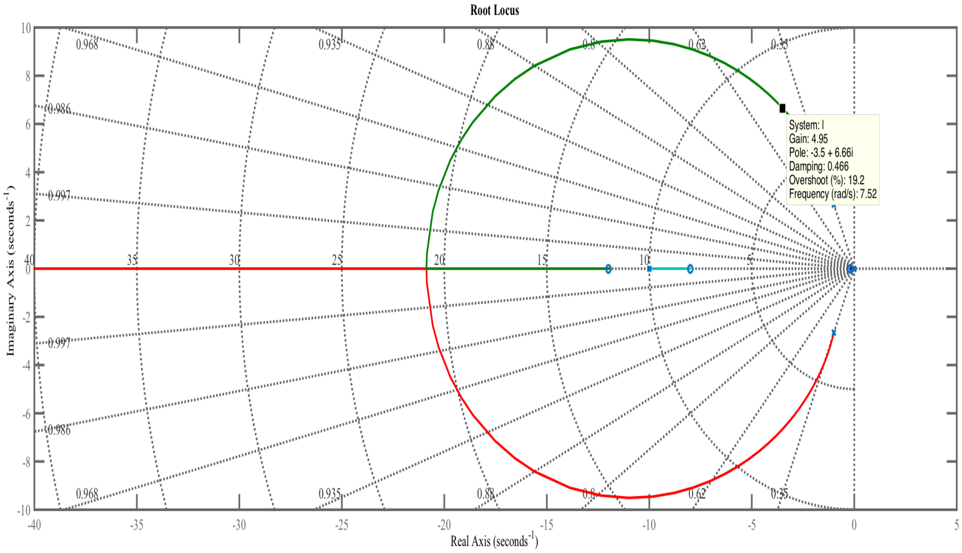

Fig. 1.5: Root Locus plot with PID Controller.

Now, the step response of the compensated system and the uncompensated system was checked :

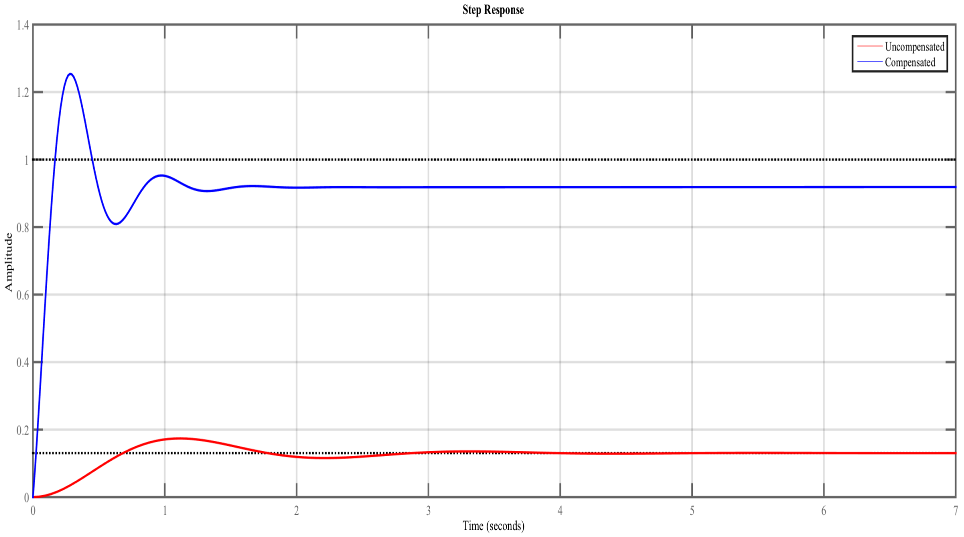

Fig. 1.6: step response of the compensated system and the uncompensated system with PID Controller.

By using code (A2.2) and by observing the figures ,then the following New Parameters were achieved :

Overshoot	19.2%
Settling Time	1/Re(pole)= 1.14
Kp	inf
Steady State Error ,ep	1\(1 + K_P)= 0
Gain	4.95

1.3 Case 3: Improving Response Using Lag Compensator.

1.3.1 Controller Physical Realization and Design .

$$ep,new= \frac{1}{kp,new1}$$

$$ep,old=\frac{1}{kp,old16+1}$$

Lag compensator:

$$\frac{kp,new}{kp,old} = \frac{zlag}{plag}$$

$$H(s) = \left(\frac{R2}{R1 + R2}\right) * \frac{s + \left(\frac{1}{R2C}\right)}{s + \frac{1}{(R1 + R2)C}}$$

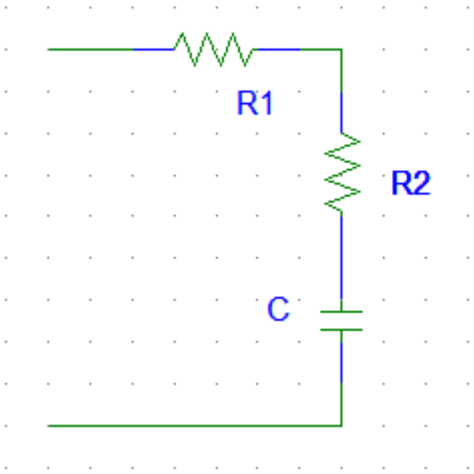

Fig. 1.7: Lag Compensator Implementation Using Passive Elements.

The root locus suggests that second order approximation is still valid since the real axis pole is far enough from the real of the dominant poles and the added zero and pole of the lag compensator almost have no effect on transient since their angular contribution is about zero and the gain is still unchanged since they contribute the same length, so they cancel each other in effect.

1.3.2 Controller Simulation.

using **Lag Compensator**, we can achieve the given specifications since the point is not located on the root locus.

At O.S =9.31%, the root locus gives the following:

Dominant second order poles: $\boldsymbol{-9.06 \mp j12}$
Gain (K) = 18

By using code (A.3.1) The root locus plot of the system was found to be :

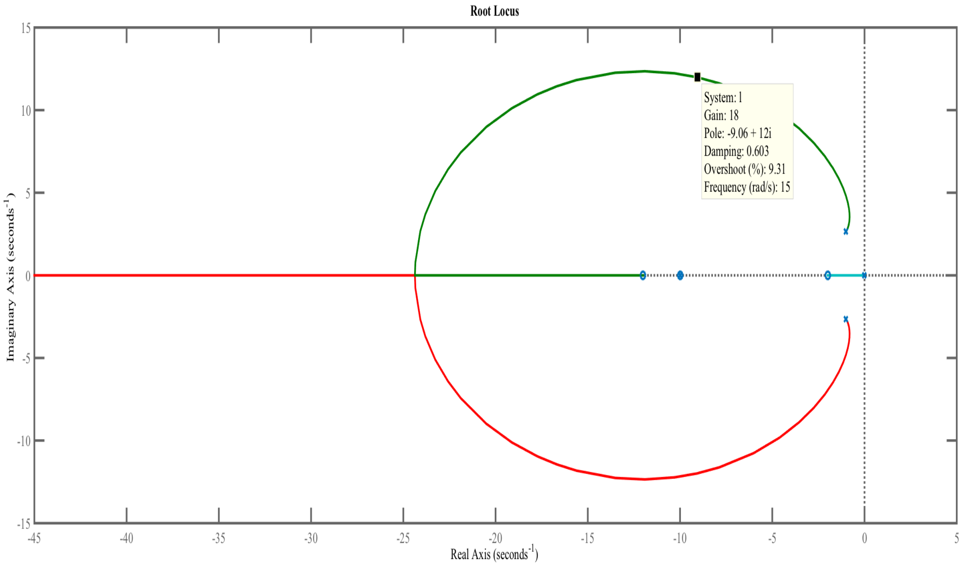

Fig. 1.8: Root Locus plot with Lag Compensator.

Now, the step response of the compensated system and the uncompensated system was checked :

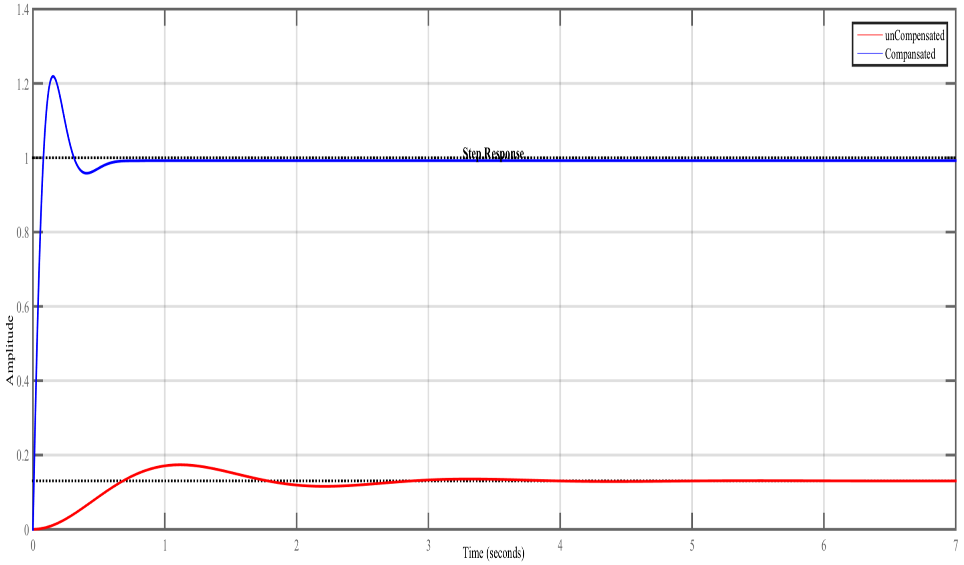

Fig. 1.9: step response of the compensated system and the uncompensated system with with Lag Compensator.

By using code (A3.2) and by observing the figures ,then the following New Parameters were achieved :

Overshoot	9.31%
Settling Time	1/Re(pole)= 0.4415
Kp	600
Steady State Error ,ep	1\(1 + K_P)= 0.0017
Gain	18

1.4 Case 4: Improving Steady State Error For System in Case 1 using PID.

1.4.1 Controller Physical Realization .

To Improve the SSE of system in Case I by 30%:

$$ep_{new}= 0.7ep_{old}$$
$$=0.7*0.0526$$
$$=0.0368$$
$$0.0368=\frac{1}{kpnew+1}$$
$$Kpnew = 1.4545$$

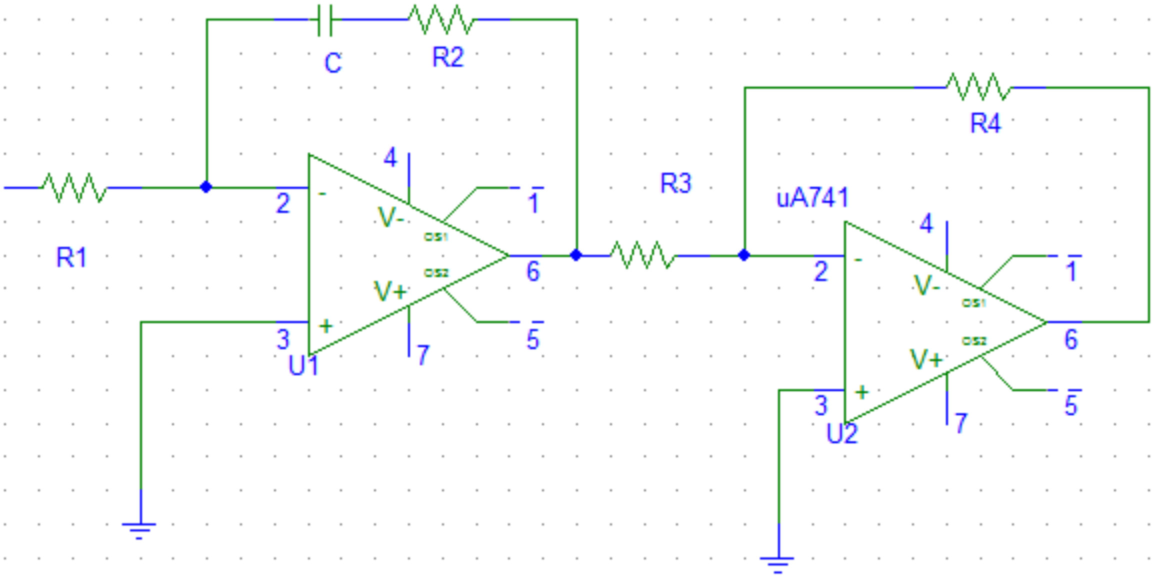

Fig. 1.10: PI Controller Implementation Using Active Elements.

$$\mathrm{Gc(s)} = \mathrm{K}\frac{S+Zc}{S}$$

$$H(s) = -\frac{R2}{R1}*\frac{s+\frac{1}{R2C}}{s}$$

$$\frac{1}{R2C} = 0.005, assume\ C = 100uF, \qquad R2 = 1Mohm$$
$$\frac{R2}{R1} = 1\,, R1 = 1Mohm$$

R4=R3= 2kohm

1.4.2 Controller Simulation.

using **PI Controller**, we can achieve the given specifications since the point is not located on the root locus.

With the same specifications of system in Case 1 but with Improves SSE :
At O.S =17.8%, Dominant second order poles: $-5.84 \mp j10.6$

By using code (A.4.1) The root locus plot of the system was found to be :

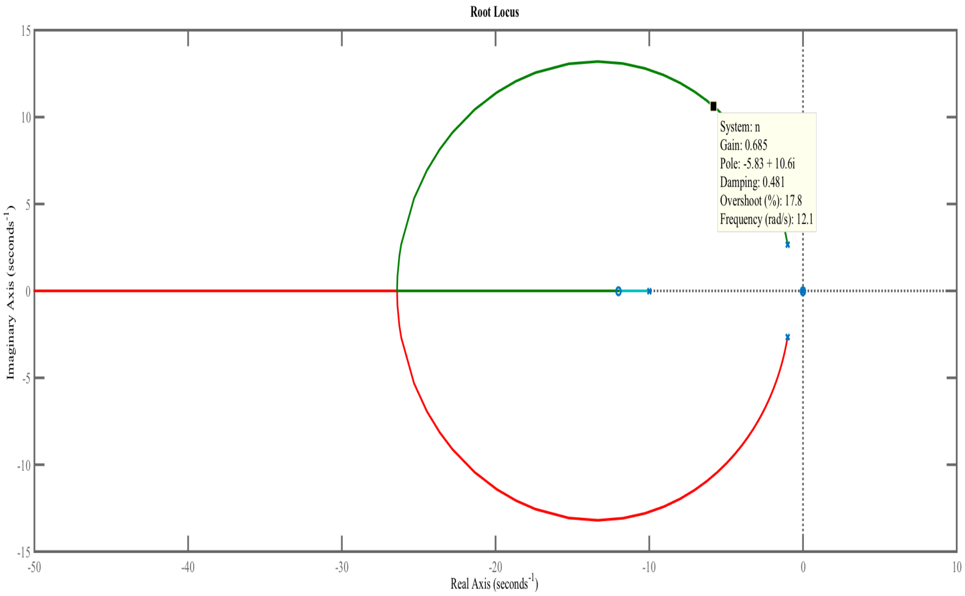

Fig. 1.11: Root Locus plot with PI Controller.

Now, the step response of the compensated system and the uncompensated system was checked :

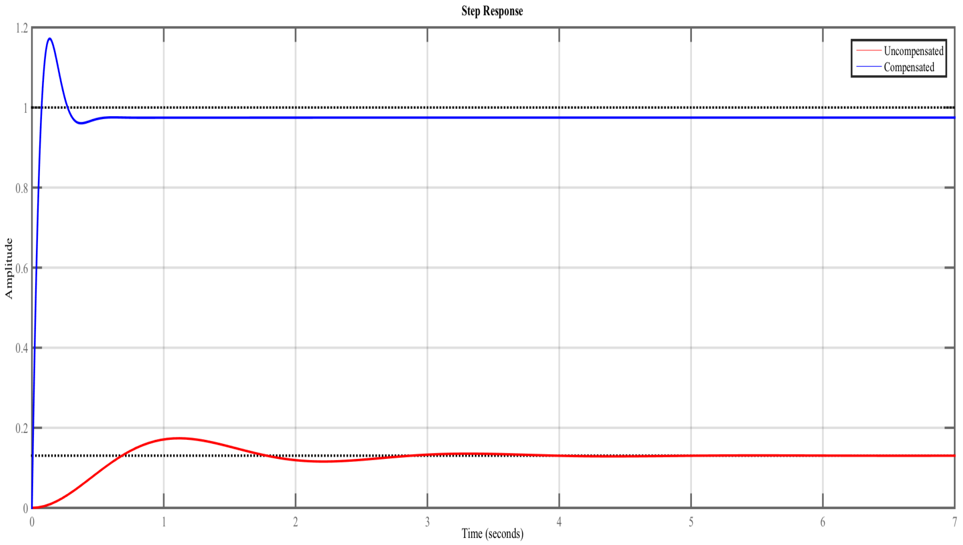

Fig. 1.12: step response of the compensated system and the uncompensated system with PI Controller.

By using code (A4.2) and by observing the figures ,then the following New Parameters were achieved :

Overshoot	17.8%
Settling Time	1/Re(pole)= 0.6826
Kp	26.2
Steady State Error ,ep	1\(1 + K_P)= 0.0368
Gain	1.454

Conclusion

At the end of this work, the design of active and passive compensators has been accomplished for different specifications on system dynamic (settling time and overshoot) and steady state error. The design was based on the root locus of the open loop system. Throughout the design, I tried to achieve the given specification under the validity of second order approximation of the compensated systems. It was shown that as the effort is made to keep a good transient response when correcting steady state error; this would be reflected to the values of components used in system realization.

References

[1] Beards C.F., Vibrations and Control System. Ellis Horwood, 1988.

[2]Bennett S., A History of Control Engineering. 1800-1930. IET. 142−148, June 1986.

Appendices

A0

```
j =[1 12 ];
M = conv([1 2 8 ],[1 10]);
G=tf(j,M);
n=[1 12 ];
d=[1 12 28 80];
poles=roots(d)
zeros=roots(n)
rlocus(G)
```

A.1.1

```
K=10;
r=tf([1 12],[1]);
l=(10*G*r);
rlocus(l)
T=feedback(G,1);
T1=feedback(K*l,1);
step(T,'r')
hold on
step(T1,'b')
```

A1.2

```
kp=dcgain(l)
ep=(1/(1+kp))
Ts=4/Re(pole)
```

A.2.1

```
K=4.95 ;
q=conv([1 8],[1 0.2]);
w=[1 0];;
r=tf(q,w);
l=(G*r);
rlocus(l)
T=feedback(G,1);
T1=feedback(K*l,1);
step(T,'r')
hold on
step(T1,'b')
```

A2.2

```
kp=dcgain(l)
ep=(1/(1+kp))
Ts=4/Re(pole)
```

A.3.1

```
K=18;
s=tf('s')
r=((s+10)*(s+2))/(s+0.005)
l=G*r;
rlocus(l)
T=feedback(G,1);
T1=feedback(K*l,1);
step(T,'r')
hold on
step(T1,'b')
```

A3.2

```
kp=dcgain(l)
ep=(1/(1+kp))
Ts=4/Re(pole)
```

A.4.1

```
K=1.454 ;
q=K*[1 0.01];
w=[1 0];;
r=tf(q,w);
n=(l*r);
rlocus(n)
T=feedback(G,1);
T1=feedback(K*l,1);
step(T,'r')
hold on
step(T1,'b')
```

A4.2

```
kp=dcgain(l)
ep=(1/(1+kp))
Ts=4/Re(pole)
```