Digital Controls Final Project 2017

Design of Classical PID and State Variable Compensator

Nilay Kant Singh, Ramachandra Vikas Chamarthi
Email: nsingh13@uncc.edu and rchamart@uncc.edu

***Abstract*—This report summarizes Digital Controls Final Project: comparison of stat-variable and classical system design approaches. In this project, a continuous-time plant function was assigned to the group, with design goals for unity feedback and 100 Hz sampling being steady-state error < 5% , settling time to within 10% of less than 0.1 second, rise time would be less than 0.1 seconds and peak overshoot to be less than 5%. The first task was to design a classical digital PID/Lag-Lead/whatever controller D(z) to meet the goals. The second task was to design a digital state-variable observer and controller with pole placement to meet the goals. It was not possible to meet the design goals using a classical D(z) controller, and the best result was steady-state error 2% and settling time to within 10% of 1.3 seconds. However, a successful digital state-variable observer and controller design achieved steady-state error 2.3% and settling time to within 10% of 0.081 seconds Theory and simulation results are provided for the two designs.**

I. Introduction

The main aim of the project is to design a Digital Control System which compares two methods of design for particular set of design parameters. Using a continuous time plant function, we needed to design a classical digital PID controller to achieve the goals and compare it with the design of a digital state variable controller and observer which uses the pole placement technique to achieve the desired goals.

The classical controller used in the project to achieve the goals is PID controller. We used PID controller [1] as our classical design method as it is easy in implementing and is feasible. PID gains can be designed based upon the system parameters and PID controller balances all the three gains which impacts the whole system which comprises of settling time, overshoot, oscillations. For state variable controller design we used Observer-Controller design as it is best suited for theoretical treatment of control systems and for numerical calculations[1]. This method also gives an idea about the internal state of the system and can be applied to multiple input multiple output systems. By comparing both the design techniques, it was found that state-variable controller performed better than classical PID controller as we design PID controller only by taking gain margin into consideration, hence just considering gain margin it gives a stable system but did not meet other requirements. But in state variable controller case we meet the requirements of the design.

In the following section, we discuss the design of classical system according to design goals. Section III provides the discussion of state-variable controller observer design to achieve the design goals. Section IV states the comparison between both the design techniques giving the details of goals that are achieved by both the design techniques. Section V gives the summary of the project giving the conclusion of what we did to achieve the goals by using both the design techniques and giving the code used and developed for both the design techniques.

II. Classical System Design

The block diagran for the digital PID controller and system is shown in Fig. 1, where sample time T_0=0.01 s digital phase-lag controller D(z), with plant function Gp(s)= 1000/(s^3 + 3 s^2 + 12 s + 10) , feedback H(s)=1/10, and a ZOH included in $Gc(s)=Gp(s)(1-e^{-sTo})/s$.

For this project, our assigned plant transfer function (including ZOH) was

$$Gc(s) = \frac{Gp(s)(1-e^{-sTo})}{s}$$

$$Gc(s) = \frac{1000(1-e^{-sTo})}{s(s^3+3s^2+12s+10)} \text{ ,(1)}$$

with corresponding G(z)

$$G(z) = \frac{0.0002753\,z^4 + 0.000548\,z^3 - 0.001622\,z^2 + 0.0005335\,z + 0.0002656}{z\text{^}5 - 3.969\,z\text{^}4 + 5.91\,z\text{^}3 - 3.912\,z\text{^}2 + 0.971\,z + 0.0002656} \text{ , (2)}$$

where Ts=0.01 s for the sampler. Since H(s)=1, hence Gol(z)=G(z).

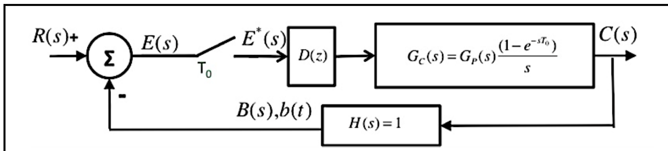

Fig. 1. **System block diagram** with T_0=0.01 s digital phase-lag controller D(z), with plant function Gc(s)= Gp(s) $(1-e^{sTo})/s$, feedback H(s)=1, and a ZOH.

The digital PID controller used in the classical system design approach is [1]

$$D(z) = Kp + Ki.\frac{z}{z-1} + \frac{Kd(z-1)}{z}$$

$$D(z) = \frac{1.664\, z^2 - 3.294\, z + 1.63}{z^2 - z}\ ,(3)$$

where $K_I = -\frac{\Omega \sin(\Theta d)}{|GcH(w1)|}$

$Kp = \frac{\cos(\Theta d2)}{|GcH(3w1)|}$

$K_D = \frac{1}{3\Omega}\left(\frac{\sin(\Theta d2)}{|GcH(3w1)|} + \frac{KI}{3\Omega}\right)$

The open-loop transfer function magnitude $|G_{OL}(z)|$ and phase $\angle G_{OL}(z)$ are shown in Fig. 2, using the digital controller D(z) above, plotted along with the uncompensated plant |G(z)| and ∠G(z) and controller plant |D(z)| and ∠D(z). As can be seen, the phase margin previously was -75.2 deg and the gain margin was -31.8dB but the phase margin now is 18.2 deg and the gain margin is 14dB

The open-loop transfer function magnitude $|G_{OL}(z)|$ and phase $\angle G_{OL}(z)$ are shown in Fig. 2, using the digital controller shown above(2). As can be seen, the phase margin is 19.7 deg and the gain margin is 15.6dB. As seen from figure the phase margin and gain margin change for compensated, uncompensated and

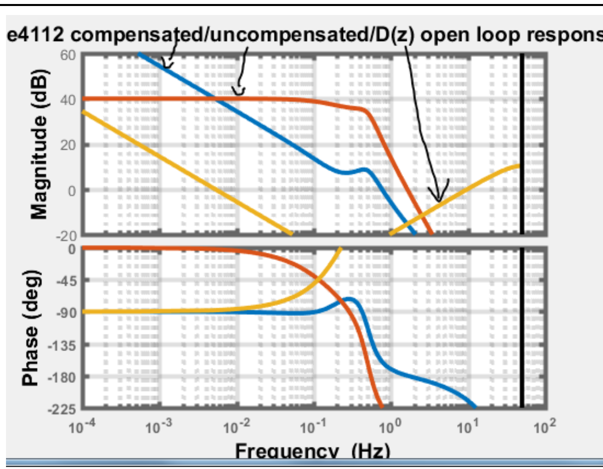

Fig. 2. Open-loop response $G_{OL}(z)$ with compensator (solid), with T_0=0.01 s using digital PID controller D(z), along with the uncompensted plant (orange) |G(z)| and ∠G(z) and compensator (yellow) |D(z)| and ∠D(z).

The corresponding closed-loop transfer function magnitude $|G_{CL}(z)|$ and phase $\angle G_{CL}(z)$ are shown in Fig. 3 along with $|G_{OL}(z)|$ and phase $\angle G_{OL}(z)$, using the digital controller inof (2). As can be seen, the 3 dB closed-loop bandwidth is 2.17Hz and the dc gain 0.0015 and the gain margin is 14dB.

From the figure below it shows the magnitude and phase of closed loop and open loop response that we obtain by using the PID controller design.

:

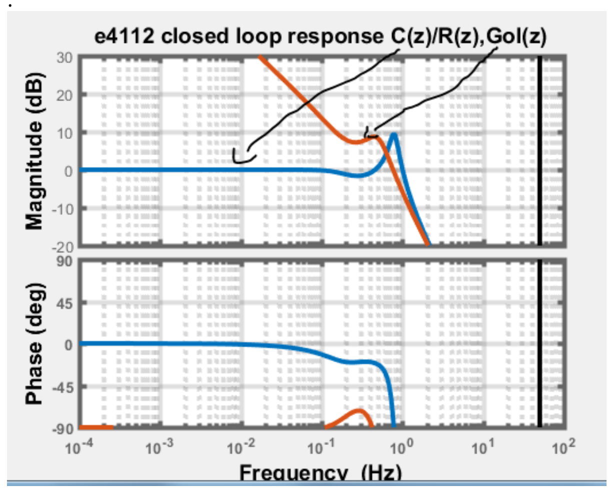

Fig. 3. Closed-loop frequency response $G_{CL}(z)$, with T_0=00.1 s digital PID controller D(z), along with open-loop response $|G_{OL}(z)|$ and phase $\angle G_{OL}(z)$. Closed-loop gain is 0.0015, and bandwidth 2.17 Hz.

A plot of the closed-loop step response is shown in is in Fig. 4. As can be seen, the settling time is 2.3 sec and the peak overshoot is 26%. **Ringing is also observed at a frequency of zzz, most likely due to a pole at zzz**. Rise time is 0.36 sec and Steady-state error was observed as less than 5%.

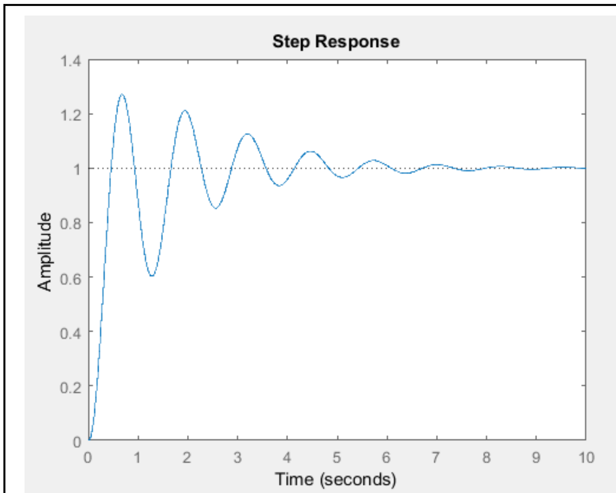

Fig. 4. Closed-loop step response for PID c(t.) Closed-loop gain is 0.015, overshoot is 26%, and rise time is 0.36 sec.

Table I lists the poles and zeroes of the closed loop system.All the poles except for one are inside the unit circle

poles of Open and Closed-Loop whatever/PID/lag-lead design

	Complex Value	Magnitude
Open-loop Poles	0.0000 + 0.0000i 0.9896 + 0.0297i 0.9896 - 0.0297i 1.0000 + 0.0000i 0.9900 + 0.0000i	0 0.9900 0.9900 1.0000 0.9900
Closed-loop Poles	0.9932 + 0.0492i 0.9932 - 0.0492i 0.9914 + 0.0077i 0.9914 - 0.0077i -0.0003 + 0.0000i	0.9944 0.9944 0.9915 0.9915 0.0003

Table II compares goals and observed performance for the PID controller design. As shown in Table II below, 2 of the 5 goals were met. The final design goal of 5% overshoot,settling time 0.1 s, rise time 0.1 s were not able to achieve as we took gain margin only in consideration because of which all the design goals were not achieved. Oscillations are due to complex conjugate poles in the system0.9896+/- 0.0497i and ringing stops once that poles are poles because of which sudden increase of phase happens causing system to stay on verge of instability.

TABLE I. DESIGN GOALS AND RESULTS

Clock Signal Name	*Design Goal*	*Final Design*
Settilng time +/- 10%	0.1 s	2.3 s
Steady-state error	5%	4 %
Overshoot peak	5 %	26 %
Rise Time	0.1 s	0.36 s
Closed-loop 3 dB bandwidth	>2 Hz	2.17

The classical Design was only made considering the phase margin and that was achieved with design thus not meeting other specifications if any one specification is made to match. If overshoot is made to match settling time was getting worse and vice versa. Intricate dependency of KP,KI and KD values didn't let us tune manually to meet the spec.

III. STATE-VARIABLE OBSERVER+CONTROLLER DESIGN

The block diagram for the state-variable observer+controller digital compensator approach is shown in Fig. 5, where sample time T_0=0.01 s, and was

$$\bar{y}[n]=\bar{\bar{C}}\bar{x}[n]+\bar{\bar{D}}\bar{u}[n]$$
$$\bar{x}[n+1]=\bar{\bar{A}}\bar{x}[n]+\bar{\bar{B}}\bar{u}[n].$$

The state variables are derived from plant function G(s), resulting in :

$$A=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -a_3 & -a_2 & -a_1\end{bmatrix}=\begin{bmatrix}2.9693 & -1.4699 & 0.4852\\ 2.0 & 0 & 0\\ 0 & 1.0 & 0\end{bmatrix} \quad B=\begin{bmatrix}0.0313\\ 0\\ 0\end{bmatrix}$$
$$C=\begin{bmatrix}b_3-b_0a_3 & b_2-b_0a_2 & b_1-b_0a_1\end{bmatrix}=\begin{bmatrix}0.0053 & 0.0105 & 0.0026\end{bmatrix}$$
$$D=[0]$$
.

The state variables above shows the values that we get by using the G(z).

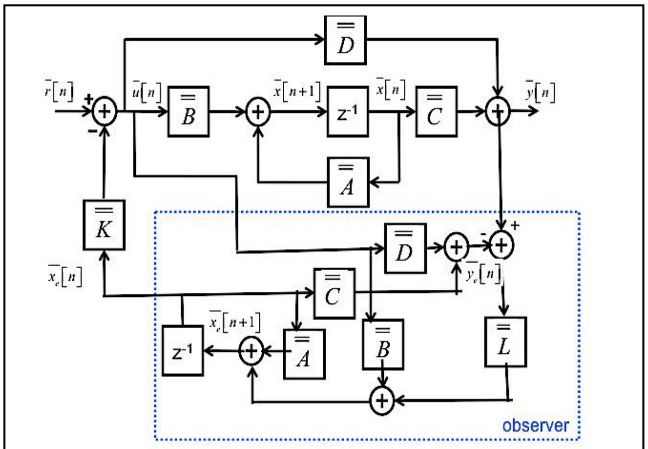

Fig. 5. State-variable observer and controller block diagram corresponding to system G_C(s) of Fig, 1

A controller K was designed for a target pole placement polynomial of p(z) having poles at z={-40.9+40.9i, -40.9-40.9i,-150}, where:

p(z)=z^3 − 1.442 z^2 + 0.7133 z − 0.09847
poles={-40.9+40.9i,-40.9-40.9i,-150} □4□
K=[48.8668 -35.6223 13.9516]

where Ts=0.01 s for the system. Poles were placed at these points based on damping ratio computed from the overshoot and settling time using the formula:-

$$€=-\frac{\ln\left(\%\frac{OS}{100}\right)}{sqrt\left(pi^2+(\ln\left(\%\frac{OS}{100}\right))^2\right)}$$

An observer was designed with observer pole placement polynomial of p2(z) having poles at z={-101, -102, -103}, where:

p2(z)= z^3 - 1.082 z^2 + 0.3901 z - 0.04689

poles={-101,-102,-103} □5□

L=[95.6253 117.1399 57.7184]

where Ts=0.01 s for the system. Poles were placed far enough such that controller poles are observable by observer and can be seen that given system tends to observer pole placed system which is our required system.

Fig. 6 shows the closed-loop frequency response for the state-variable digital observer+controller design, along with the losed-loop response of the classical design of the previous section. The state-variable design greatly reduced the peaking in the response while also improving bandwidth. The peaking in bodeplot of PID compensator was responsible the high overshoot and high ringing in the system and that can be seen made flat in the State Observer Design.

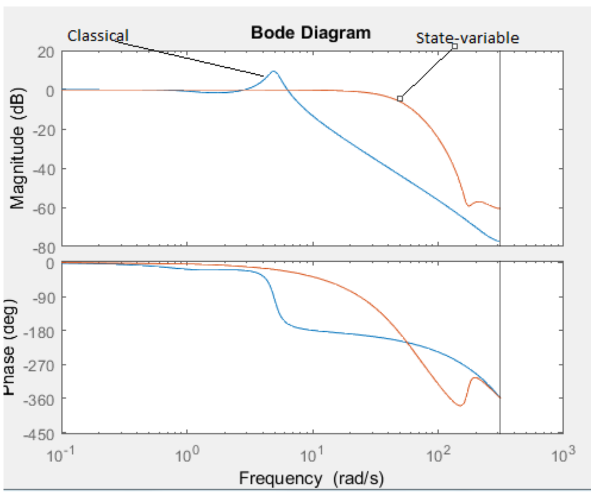

Fig. 6. Closed-loop frequency response for state-variable design (orange) and classical design (blue), with T_0=0.01.

Fig. 7 shows the closed-loop step response for the state-variable digital observer+controller compensated design, along with the closed-loop response of the classical design of the previous section. The state-variable design greatly reduced the peaking in the response while also improving bandwidth..

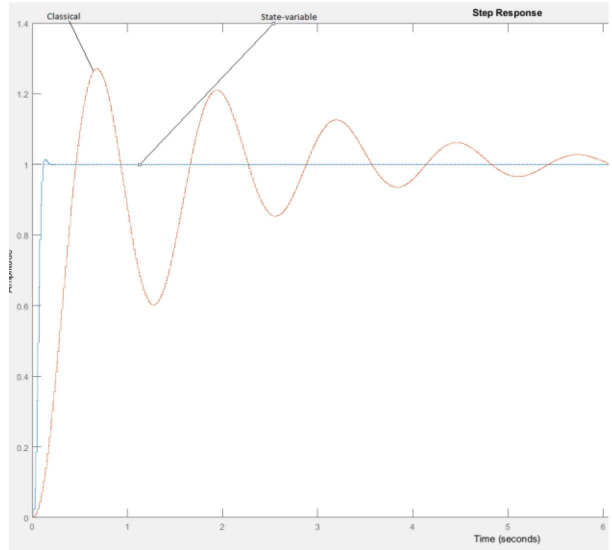

Fig. 7. Closed-loop step response for state-variable design (blue) and classical design (orange), with T_0=0.01.

Table III lists the poles and zeroes of the closed-loop compensated observer+controller state-variabe system. All of the open-loop have magnitude less than 1 and are in stable region causing a stable system and above rresponse can also be seen stable and also meeting all the specification

TABLE II.

TABLE III. POLES OF CLOSED-LOOP STATE-VARIABLE DESIGN

	Complex Value	Magnitude
Closed-loop Poles	0.6095 + 0.2642i 0.6095 - 0.2642i 0.3642 + 0.0000i 0.3606 + 0.0000i 0.3570 + 0.0000i 0.2231 + 0.0000i	0.6643 0.6643 0.3642 0.3606 0.3570 0.2231

IV. COMPARISONS OF BOTH METHODS TO DESIGN GOALS

As shown in Table IV below, 4 goals were met using the state variable controller, while 2 goals were not met using the classical PID controller. The final design goal bandwidth was not met, because the bandwidth calculation was going toward zero making the 3dB bandwidth to be low.

Table IV compares goals of classical PID controller design and state-variable controller design giving us a clear view of that the state-variable method is more efficient in designing digital control systems for particular set of design requirements and can meet them easily as compared to classical design method.

TABLE IV. DESIGN GOALS AND RESULTS FOR BOTH METHODS

Clock Signal Name	*Design Goal*	*Classical Design*	*State Variable*
Settilng time +/- 10%	0.1 s	2.3 s	0.1 s
Steady-state error	5%	4 %	4%

Clock Signal Name	*Design Goal*	*Classical Design*	*State Variable*
Overshoot peak	5 %	26 %	3 %
Rise Time	0.1 s	0.36 s	0.1 s
Closed-loop 3 dB bandwidth	>2 Hz	2.17 Hz	1 Hz

In addition, state-variable is preferred over classical PID controller design method as it is flexible and easy to implement even the difficult systems.

V. Summary

Two digital controllers were designed, state –variable and classical PID controller for a particular system with specific design requirements. Classical method did not achieve all the design requirements whereas state-variable almost achieved all the design goals.

Reasons why we failed in designing PID controller with specific design goals was that we took only gain margin into consideration for calculating the three different Kp,Ki,Kd value which should be tuned in such a way that all the design goals were not able to be achieved.

Reasons why we succeeded in designing state variable controller with specific design requirements was that it checked the system for controllability and observability using the pole placement technique which helped the observer to observe the controller poles as they were wide apart from each other. This system helped to achieve almost all the design goals thereby stating that state-variable controller method is better than classical PID controller method for designing complex digital control systems.

References

[1] Phillips and Nagle and Chakrabortty, "Digital Control System Analysis and Design", 4th edition Pearson Publication,2014. *and Design", 4th edition Pearson Publication,2014.*

[2] T.P. Weldon, J.M.C. Covington III, K.L. Smith, and R.S. Adams ``Performance of Digital Discrete-Time Implementations of Non-Foster Circuit Elements," *2015 IEEE Int. Sym. on Circuits and Systems*, Lisbon, Portugal, May 24-27, 2015.

[3] T.P. Weldon, J.M.C. Covington III, K.L. Smith, and R.S. Adams, ``Stability Conditions for a Digital Discrete-Time Non-Foster Circuit Element," *2015 IEEE Int. Symposium on Antennas and Propagation*, Vancouver, BC, Canada, July 19-25, 2015.

[4] Won Namgoong An observer-controller digital PLL — A time-domain approach, Radio and Wireless Symposium (RWS), 2016 IEEE.

Appendix 1

Mattlab Script

Classical PID controller:-

```
%% E4112 Project 7 Matlab
%% copyright 2017 by Thomas Weldon

close all
clear all

Ts = 0.01;  %sample period
fmax=100 ;   %max freq for plotting
fmin =fmax/1000000;
zz=[    ]';
pp=[   -1+3i -1-3i  -1     ]';
kk=10^3       %*0.01*110.12/5.40754
[tfnum,tfden] = zp2tf(zz,pp,kk)
Gp.s = tf(tfnum,tfden);   %plant function
%Gp.s = tf([10000],[1 70 1000 0]);   %plant function 10,000/s(s+20)(s+50)
Gp.z = c2d(Gp.s, Ts, 'zoh');
Gp.z
st = genss(Gp.z)
ss(st)
H.s = tf([1],[1]);           %feedback continuous-time filter
H.z = c2d(H.s, Ts, 'zoh');
H.z



%************************************************
% enter your PID design coefficients here
kp=0.079796*0.43;  ki=0.077077*0.43;  kd=0.0379*0.43;
% kp=1;  ki=0;  kd=0;
pidCtrl=pid(kp,ki,kd,0,Ts,'IFormula','Trapezoidal','DFormula','BackwardEuler');
pid.z=tf(pidCtrl);
D.z=pid.z
% D.z = 1
D.s = d2c(D.z,'Tustin');
Gfwd.z = series(D.z,Gp.z);
Gfwd.s = series(D.s,Gp.s);
Gol.z = series(Gfwd.z,H.z);%compensated open loop response
GolUncomp.z=series(Gp.z,H.z);%uncompensated open loop response
Gol.s = series(Gfwd.s,H.s);
Gcl.z = feedback(Gfwd.z, H.z); %closed loop pulse trans func
Gcl.s = feedback(Gfwd.s, H.s);

figure(1);
set(gca,'linewidth',3.0);
aa=gca
aa.XGrid='on';aa.YGrid='on';aa.LineWidth=3;aa.FontWeight='bold';
aa.XLabel.FontSize=20;aa.YLim=[-225 0];
set(findall(gcf,'Type','line'),'LineWidth',3);
```

```
set(findall(gca,'Type','line'),'LineWidth',3);
set(findall(gcf,'Type','text'),'FontSize',14);
set(findall(gcf,'Type','text'),'FontWeight','bold');
bplt=bodeplot(Gol.z,GolUncomp.z,D.z)
setoptions(bplt,'FreqUnits','Hz','grid','on','YLim',[-20 60],'XLim',[fmin  fmax]);
title('e4112 compensated/uncompensated/D(z) open loop response')
%legend('C(z)/R(z)','Location','northwest')
aa=gca
aa.XGrid='on';aa.YGrid='on';aa.LineWidth=3;aa.FontWeight='bold';
aa.XLabel.FontSize=20;aa.YLim=[-225 0];
set(findall(gcf,'Type','line'),'LineWidth',3);
set(findall(gca,'Type','line'),'LineWidth',3);
set(findall(gcf,'Type','text'),'FontSize',14);
set(findall(gcf,'Type','text'),'FontWeight','bold');

figure(2);
set(gca,'linewidth',3.0);
bplt=bodeplot(Gcl.z,Gol.z)

setoptions(bplt,'FreqUnits','Hz','grid','on','YLim',[-20 30],'XLim',[fmin  fmax]);
title('e4112 closed loop response C(z)/R(z),Gol(z)')
%legend('C(z)/R(z)','Location','northwest')
aa=gca
aa.XGrid='on';aa.YGrid='on';aa.LineWidth=3;aa.FontWeight='bold';
aa.XLabel.FontSize=20;aa.YLim=[-90 90];
set(findall(gcf,'Type','line'),'LineWidth',3);
set(findall(gca,'Type','line'),'LineWidth',3);
set(findall(gcf,'Type','text'),'FontSize',14);
set(findall(gcf,'Type','text'),'FontWeight','bold');

figure(3);
set(gca,'linewidth',3.0);
%zplane(Z,P) plots the zeroes Z and poles P (in column vectors) with the
pzmap(Gcl.z);
title('Closed loop response C(z)/R(z) poles and zeroes')
axis([-1.5 1.5 -1.5 1.5]);
%zeta = [ 0.25 0.5 0.75];
Wn = [0.25 0.5 0.75]*3;
aa=gca
aa.XGrid='on';aa.YGrid='on';aa.LineWidth=3;aa.FontWeight='bold';
aa.XLabel.FontSize=20;aa.FontSize=12;
set(findall(gcf,'Type','line'),'LineWidth',3);
set(findall(gcf,'Type','text'),'FontSize',14);
set(findall(gcf,'Type','text'),'FontWeight','bold');
figure(4);
margin(Gol.z)
figure(5);
step(Gcl.z)
figure(6);
margin(GolUncomp.z)
figure(7);
margin(Gcl.z)
```

State-Variable controller:-

```
clc
clear all
close all
%Pole locations of controller
p1 = -40.9 + 40.9i;
p2 = -40.9 - 40.9i;
p3 = -150;
%Pole locations of Observer
p4 = -101
p5= -102
p6 = -103
zz=[]
pp=[   -1+3i -1-3i  -1     ]';
kk=10^3;
Ts = 0.01
[tfn,tfm]=zp2tf(zz,[p1 p2 p3],1)
h = tf(tfn,tfm)
h = c2d(h,Ts,'zoh')
poles1=pole(h)
[tfn,tfm] = zp2tf(zz,[p4 p5 p6],1)
i = tf(tfn,tfm)
i = c2d(i,Ts,'zoh')
poles2 = pole(i)
[tfn,tfm]=zp2tf(zz,pp,kk)
Gp.s = tf(tfn,tfm);

G.z=c2d(Gp.s,Ts,'zoh')
sv = ss(G.z);

C = ctrb(sv.A,sv.B)
O = obsv(sv.A,sv.C)
u=rank(C)
if u==3
    disp('Controllable')
end

v=rank(O)
if v==3
    disp('observable')
end
K = place(sv.A,sv.B,poles1)
L = place(sv.A',sv.C',poles2)
Rg=reg(sv,K,L')
Feed=feedback(sv,Rg,1)


I =eye(size(sv.A))
G = (K*inv(I-(sv.A-sv.B*K-L'*sv.C)))*L'

step(G*Feed)
```