Advanced Control System

Term Project

Submitted by:

Name: Aniket Kumar

Roll no: 200108068

Group no: 16

Date: 19 /04/2023

Course Instructor:

Dr. Parijat Bhowmick

Question 1:

Design a Compensator for unity feedback system with open loop transfer function G(s) = K /(K/(s(s+1)(0.5s +1)) to satisfy the following specifications:

(i) Velocity error constant K_v=5
(ii) Phase margin =40 degrees.
(iii) Gain Margin = 10dB

Code:

```
%% Advanced Control System
% Submitted by
% Aniket Kumar
% 200108068
% Group 16
% Question 1
%% Design of Lag Compensator
% Design a Compensator for a unity feedback system with
% open loop transfer function G(s)= 1/(s(s+1)(0.5s +1)) to satisfy
the
% following specifications:
% (i) velocity error constant K_v=5,
% (ii) Phase margin =40 degrees.
% (iii)Gain Margin = 10dB
close all
clear all
clc
% Adjusting the open loop transfer function of K_V = 5
% So , G1 = k*G
% Kv = lim(s->0) sG1(s)H(s) = 5
% Kv = lim(s->0) s*K/(s(s+1)(0.5s +1))*1 = 5
% Therefore Kv = K = 5
% Uncompensated Sys = 5/(s(s+1)(0.5s +1))
```

```
num1 = [5];
den1 = [0.5 1.5 1 0] ;
disp(['Uncompensated system Transfer function']);
G1 = tf(num1 , den1 ) ;
% printing to console the uncompensated Tf with adjusted Kv
G1
%Extracting the Gm , Pm of G1
figure
grid
[Gm, Pm, w_gm, w_pm] = margin(G1);
margin(G1)
title({"Uncompensated system with gain adjusted Kv=5sec^-
1",sprintf('Phase Margin = %0.4f deg, Gain Margin = %0.4fdB',Pm,
Gm)})
%% Lag Compensator
phie_req = 40 ;
tolerance = 5;
phie_req = 40 +tolerance  ;
% Noting down the W value at 180 - phir_req i.e 180 -45 = 135 from
the
% Bode Plot of G1
w_at_135 =  0.554;
% Noting Down the Gain at W = 0.554 from bode plot of G1
% Gain = 17.5
g_un_w = 17.5;
% We then design a lag compensator and we know that lag compensator
brings down
% the magintude plot by beta such that the phase crossover
frequency
% becomes wc/10.
```

```
% Lag Compensator = (sT +1) / (sBT +1 ) where B>1
% we know , 20 log(B) = g_un_w
beta = 10 ^(17.5/20);
wc_z = w_at_135/10 ;
T = 1/wc_z;
wcp = 1/(beta * T) ;

% Forming the Lag Compensator using the obtained Values
numc = [T 1];
denc = [beta*T 1] ;
GC = tf(numc , denc) ;
disp(['Compensator Transfer function']);
GC
figure
% Displaying the compensator thus obtanined
margin(GC)
%% Final Compensated Transfer Function
Geq = G1 * GC;

figure
margin(Geq)

disp(['Compensator Transfer function']);
Geq

%% Ploting step response of compensated and uncompensated system
CloseLoopG1 = feedback(G1*1 ,1) ; % H(s) = 1
CloseLoopGeq = feedback(Geq*1,1) ; % H(s) = 1
figure

step(CloseLoopGeq)
```

```
title({"Step Response of  Compensated close loop system"})

figure
step(CloseLoopG1)
title({"Step Response of  Uncompensated close loop system"})
%% Ramp Response of Compensated and UnCompensated System

% Define the input signal as a ramp
t = 0:0.1:10;
alpha = 1;
ramp = alpha * t;
% Plotting for Compensated System
figure
plot(t, ramp)
hold on
lsim(CloseLoopGeq, ramp, t)
title({"Ramp Response of Compensated System"})
legend('Ramp signal','Compensated system')

hold off
% Plotting for Uncompensated System
figure
plot(t, ramp)
hold on
lsim(CloseLoopG1, ramp, t)
title({"Ramp Response of Uncompensated System"})
legend('Ramp signal','Uncompensated system')

hold off
```

Question 2:

Design a PID controller, obtain unit step response with MATLAB/ Simulink, and compare the responses of systems.
G(s) = 1 / (s) *(s+1) *(s+5)

Code:

```
%% Advanced Control System
% Submitted by
% Aniket Kumar
% 200108068
% Group 16
% Question 2 part b
%% Design a PID controller, obtain unit step response with
MATLAB/ Simulink, and compare the responses of systems
% G(s) = 1 / (s)(s+1)(s+5)
close all
clear all
clc

num1 = [1];
den1 = [1 6 5 0] ;

G1 = tf(num1 , den1 ) ;
%plotting root locus to calculate the gain and frequency at
which the closed-loop system becomes marginally stable
% (i.e., the gain and frequency at the point where the root
locus intersects the imaginary axis)
```

```
figure(1)
rlocus(G1)

% k = 30;
% w = 2.24;
% Close Loop feedback without controller



k_u = 30; % ultimate gain
w_u = 2.24; %ultimate period
pu = 2* 3.14/w_u ;
%pu = 2pie / w_u

%% Using The Ziegler-Nichols Tuning Rule | Ultimate Gain &
Ultimate Period
kp = k_u/1.7;
ki = kp/(pu/2) ;
kd = kp * (pu/8);

%% pid controller without tuning

PID_CONT1 = pid(kp , ki, kd);
PID_CONT1
open_sys = series(PID_CONT1 , G1) ;
closeSys1 = feedback(open_sys,1) ;


%% pid controller with tuning
% Fix one quantity and change other
% General trend to better the following parameters are listed
below
%% Settling time : Kp -> Small Change, Ki -> Increase, Kd ->
Decrease
%% Rise Time : Kp -> Decrease, Ki -> Decrease, Kd -> Small
Change
%% Overshoot : Kp -> Increase, Ki -> Increase, Kd -> Decrease
```

```
%% SS error : Kp -> Decrease, Ki -> Estimate, Kd -> Small
Change
delta1 = 2.8;
delta2 = 3.1;
delta3 = 1.5;
kp = k_u/1.7 - delta3;
ki = kp/(pu/2) - delta1 ;
kd = kp * (pu/8) +delta2;
PID_CONT2 = pid(kp , ki, kd);
PID_CONT2
open_sys = series(PID_CONT2 , G1) ;
closeSys2 = feedback(open_sys,1) ;

%% Plotting the step response
figure(2)
step(closeSys1)
hold on
step(closeSys2)
hold on
gcl = feedback(G1 , 1) ;
step(gcl)
legend('PID without tuning','PID with tuning','Without
Controller')
hold off
%% Details of both the responses
disp(['Step Response in PID'])
S1= stepinfo(closeSys1)
disp(['Step Response in PID further tuned'])
S = stepinfo(closeSys2)
%%%%Thank You %%%%%%
```