The University of Sheffield
ACS6101 Foundations of Control Systems
Week 5 Assignment

Paulo Roberto Loma Marconi
prlomarconi1@sheffield.ac.uk

November 4, 2018

1 Question 1

The aim of this task is to design a digital controlled system with some requirements, small steady state error, small settling time, minimum input action, and minimum overshoot.

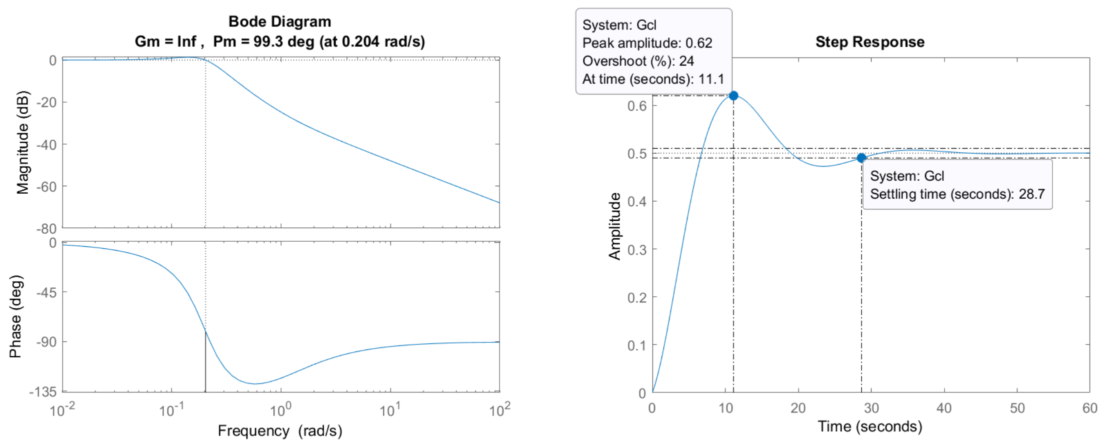
The plant to be studied is written as follows,

$$G_p(s) = \frac{0.04(s+1)}{s^2+0.2s+0.04} \tag{1}$$

and the digital controller should have the form,

$$D(z) = K\frac{z-A}{z-B} \tag{2}$$

Fig.1 shows that the plant is very slow and with a relatively big overshoot. Therefore, a phase margin (PM) of 60° can be the unique requirement. As long as the compensated phase margin is around that value, the settling time and overshoot should be minimized as much as possible.



(a) Bode plot

(b) Step response.

Figure 1: Evaluation of the plant Gp

1.1 Phase-Lead compensator

The selected compensator can be written as follows,

$$G_c = K_c \frac{s+z}{s+p}, \quad |z| \leq |p| \tag{3}$$

Step 1. Calculate a gain K that satisfies the desired phase margin $PM_d = 60°$. Applying the angle condition for a PM_d over G_p,

$$\begin{aligned} \measuredangle G_p(j\omega_c') &= PM_d - 180° \\ \measuredangle G_p(j\omega_c') &= 60° - 180° \\ \measuredangle G_p(j\omega_c') &= -120° \end{aligned} \tag{4}$$

where ω_c' is the new crossover frequency for the desired phase margin $PM_d = 60°$.

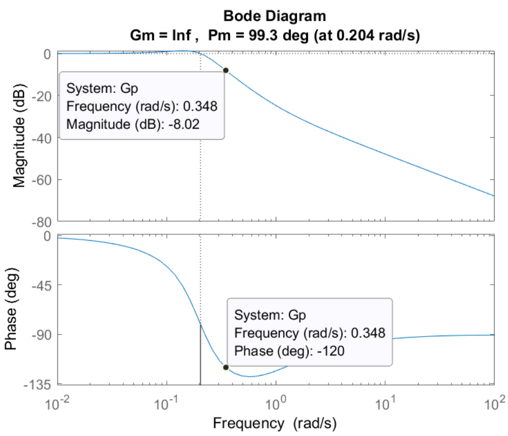
Using the bode plot of G_p, Fig. 2a, the logarithm gain at $-120°$ is $-8.02\ dB$, so the gain K can be calculated as follows,

$$\begin{aligned} 20\log_{10} K &= |-8.02| \\ K &= 2.52 \end{aligned} \tag{5}$$

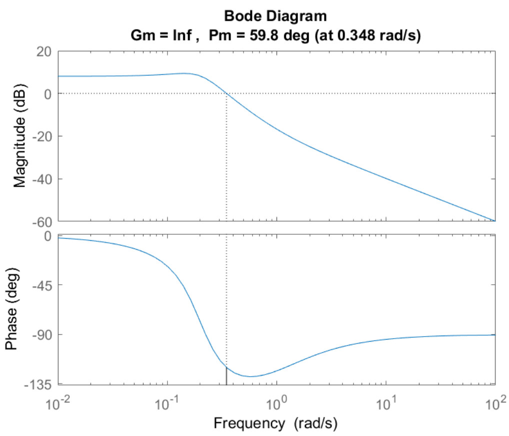
Therefore, the uncompensated G_p that satisfies the desired phase margin is,

$$\begin{aligned} G_{p1} &= K\ G_p \\ G_{p1} &= 2.52 \frac{0.04(s+1)}{s^2+0.2s+0.04} \end{aligned} \tag{6}$$

and Fig. 2b shows that the new uncompensated plant G_{p1} satisfies the desired phase margin.



(a) Bode plot of Gp



(b) Bode plot of Gp1

Figure 2: Evaluation of the plant Gp and Gp1

The following Matlab scripts simulates the previous results.

```
%% ==========================================================================
% ---- ACS6101 Assignment week 5
% ---- Registration number: 180123717
% ---- Name: Paulo Roberto Loma Marconi
% ---- 03/11/2018
%% === Question 1 ==========================================================
clear; clc; close all; % clean previous data in order to avoid erros.
%% plant Gp
```

```
s = tf('s');
Gp = 0.04*(s+1)/(s^2+0.2*s+0.04); % uncompensated plant

fig = figure(1);
margin(Gp); % calculates the phase margin and gain margin at their frequencies
[Gm,Pm,Wcg,Wcp] = margin(Gp);
Gcl = feedback(Gp,1); % closed-loop of the uncompensated plant
saveas(fig,'Q1_Gp_margin.png');

fig = figure(2);
step(Gcl); % step response to the closed-loop system
stepinfo(Gcl) % system performance values
saveas(fig,'Q1_Gp_step.png');
%% Design requirements
PO = 10; % percentage overshoot
zeta = log(100/PO)/sqrt(pi^2+ (log(100/PO))^2 ); % damping ratio
PM_d = round(100*zeta)+1; % desired PM
%% Obtaining the gain K that meets the desired PM_d
K = 10^(8.03/20); % the gain 8.03 obtained from the bode plot
Gp1 = K*Gp; % new uncompensated plant

fig = figure(11);
margin(Gp1);
saveas(fig,'Q1_Gp1_K_margin.png');
```

Step 2. The digital uncompensated G_{z1} plant of G_{p1} can be calculated using a zero-order holder with a sampling time $T_s = 0.01$.

$$G_{z1} = 1.01 \cdot 10^{-3} \frac{z - 0.99}{z^2 - 1.99z + 0.99} \tag{7}$$

In Fig. 3 it can be seen that the continuous and discrete plant are almost similar. Also, the settling time has been reduced but the steady state error is too big.

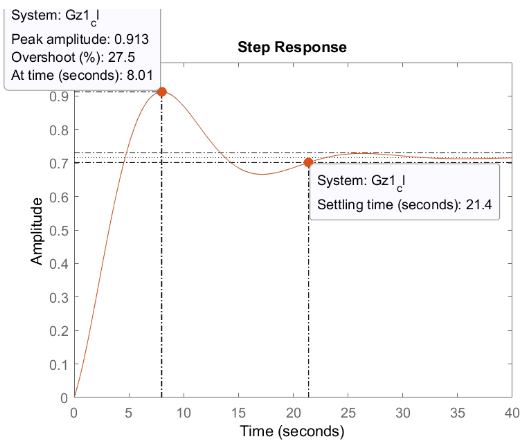


Figure 3: Step response of Gz1 and Gp1

Step 3. With the desired phase margin, the value of β can be calculated as follows,

$$PM_{act} - PM_d + \theta = \arctan \frac{\beta - 1}{2\sqrt{\beta}} \tag{8}$$

where PM_{act} is the actual phase margin of the uncompensated plant G_{p1}, and θ is a factor of correction.

After some operations,

$$\beta^2 - \beta \left[2 + 4 \left(\tan(PM_d - PM_{act} + \theta)\right)\right] + 1 = 0$$

if $\theta = 6°$, $PM_d = 60°$, and $PM_{act} = 61.25°$ obtained from Fig. 4, β will be,

$$\beta = 1.18$$

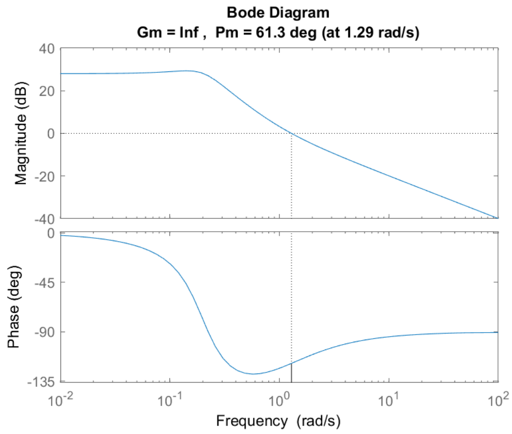


Figure 4: Bode plot of Gp2

Step 4. Now, the crossover over frequency ω_c needs to be calculated using the following gain condition formula,

$$|G_{p1}(j\omega_c)| = \frac{1}{\sqrt{\beta}} \tag{9}$$

if we use the peak magnitude M_{pc} relation,

$$M_{pc} = \frac{1}{\sqrt{\beta}} \tag{10}$$

and using `getGainCrossover(Gp2,Mpc)` on Matlab,

$$\omega_c = 1.37\ rad/sec$$

Step 5. Determining the zero of the controller,

$$\begin{aligned} \omega_c &= \sqrt{\beta z^2} \\ z &= 1.26 \end{aligned} \tag{11}$$

therefore, the compensator in continuous time can be written as follows,

$$\begin{aligned} G_c &= \beta \frac{s+z}{s+\beta z} \\ G_c &= 1.18\ \frac{s+1.26}{s+1.49} \end{aligned} \tag{12}$$

and in discrete time,

$$G_c z = 1.18\ \frac{z-0.99}{z-0.98}$$

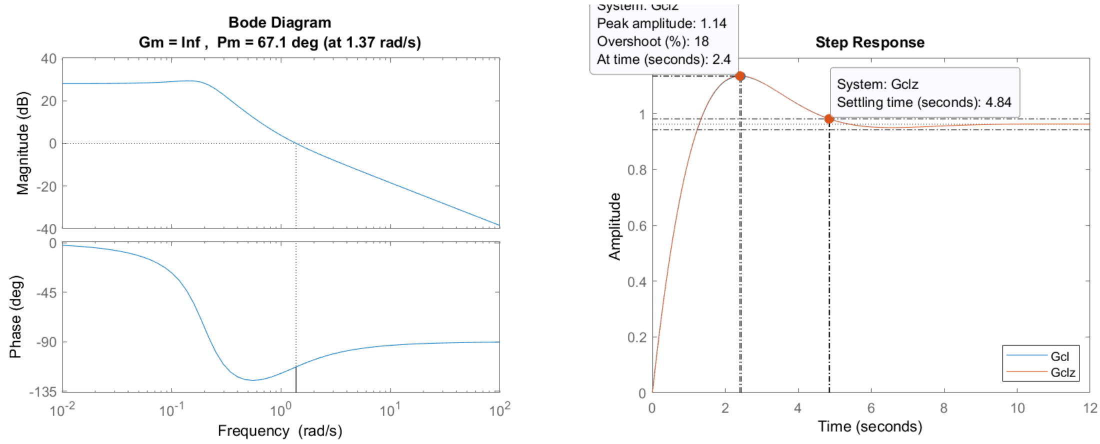
The open-loop compensated in continuous and discrete time are,

$$G_{ol} = G_c \; G_{p1} \tag{13}$$

$$G_{ol} = 1.18 \quad \frac{s+1.26}{s+1.49} \quad 2.52 \quad \frac{0.04(s+1)}{s^2+0.2s+0.04} \tag{14}$$

$$G_{olz} = 0.01 \quad \frac{z-0.99}{z-0.98} \quad \frac{z-0.98}{z^2-1.99+0.99} \tag{15}$$

Fig. 5 shows that the compensated system using a Phase-Lead controller achieves with success the desired phase margin of 60°, and has a small settling time $t_s = 4.84\ sec$.



(a) Bode plot

(b) Step response.

Figure 5: Evaluation of Phase-Lead compensated system in continuous and discrete time.

The following Matlab script simulates and evaluates the previous design.

```
%% Designing the Phase-lead digital controller
% introducing a new gain 10 times faster in ordert to obtain a fast
% response to the step input
Gp2 = K*10*Gp;

fig = figure(4);
margin(Gp2); % checking the desired phase margin PM_d
[Gm1,PM_act,Wcg1,Wcp1] = margin(Gp2);
saveas(fig,'Q1_Gp2_margin.png');

% step 1) obtaining beta with the actual PM and the desired PM
theta = 6; % correction factor
beta = roots( [1 -(2+4*( tand(PM_d-PM_act+theta) )^2) 1] );
% beta = (1+sind(PM_d-PM_act+theta))/(1-sind(PM_d-PM_act+theta));
% step 2) calculating the new crossover frequency
Mpc = 1/sqrt(beta(1)); % find compensator peak magnitude.
omega_c = getGainCrossover(Gp2,Mpc); % The new gain crossover frequency wc
% step 3) determining the zero and the pole of the controller
zc = omega_c/sqrt(beta(1)); % zero of the controller
pc = beta(1)*zc; % pole of the controller
% step 4) controller in continuous and discrete time
Gc = beta(1)*(s+zc)/(s+pc); % phase-lead controller in continuous time
Gcz = c2d(Gc,Ts,'zoh'); % phase-lead controller in discrete time
%% Evaluating the phase-lead controller
Gol = Gc*Gp2; % open-loop compensated in continuous time
Gcl = feedback(Gol,1); % closed-loop in continuous time
Gp2z = c2d(Gp2,Ts,'zoh'); % plant in discrete time
Golz = Gcz*Gp2z; % open-loop compensated in discrete time
```

```
Gclz = feedback(Golz,1); % closed-loop in discrete time

fig = figure(5);
margin(Gol);
saveas(fig,'Q1_lead_margin.png');

fig = figure(6);
step(Gcl,Gclz);
saveas(fig,'Q1_lead_step.png');
```

2 Question 2

The deadbeat controller is a type of controller that reaches zero error very fast at the sampling instants in the discrete domain systems, but it can not be used in continuous time, [1]. Another characteristics are that the control signal is very high, and the overshoot is less than 2%.

As long as the plant $G(z)$ has all zeros and poles inside the unit circle (minimum phase), the plant can be written as,

$$G(z) = \frac{z^d B(z)}{A(z)} = \frac{z^d(b_o + b_1 z^{-1} + ... + b_m z^{-m})}{1 + a_1 z^{-1} + ... + a_n z^{-n}}, \quad d = n - m > 0 \tag{16}$$

the digital deadbeat will have the following transfer function,

$$D(z) = \frac{A(z)}{z^{-d}B(z)} \frac{z^{-d}}{1 - z^{-d}} \tag{17}$$

and the closed-loop system is,

$$Y(z) = z^{-d}U(z) \tag{18}$$

The aim of this question is to design a digital deadbeat controller for the plant,

$$G_p(s) = \frac{0.04(s+1)}{s^2 + 0.2s + 0.04} \tag{19}$$

$$G_p(z) = \frac{5.47 \cdot 10^{-2}(z - 0.34)}{z^2 - 1.78z + 0.82} \tag{20}$$

with a the sampling time $T_s = 1$.

After using the following code in Matlab.

```
%% ==================================================================
% ---- ACS6101 Assignment week 5
% ---- Registration number: 180123717
% ---- Name: Paulo Roberto Loma Marconi
% ---- 03/11/2018
%% === Question 2 ===================================================
clear; clc; close all;
s = tf('s');
Gp = 0.04*(s+1)/(s^2+0.2*s+0.04); % uncompensated plant
Ts = 1; % sampling time
Gz = c2d(Gp,Ts,'zoh'); % System transfer function in discrete time
% Minimum phase case
z=zpk('z',Ts);
Mz = 1/z; % closed-loop because the relative order d = 1
Dz = Mz/(Gz*(1-Mz)); % deadbeat controller
Dz = minreal(Dz); % cancel common factors
fig = figure(1);
step(Dz*Gz/(1+Dz*Gz)) % step response of closed-loop
saveas(fig,'Q2_deadbeat_step.png');
fig = figure(2);
step(Dz/(1+Dz*Gz)) % control signal
saveas(fig,'Q2_deadbeat_control.png');
```

The closed-loop discrete system is as follows,

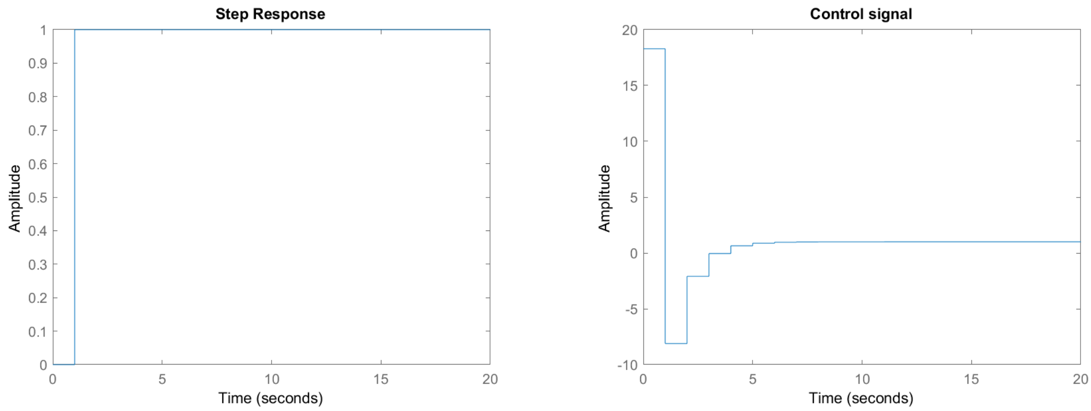
$$G_{clz} = \frac{(z-1)(z-0.3392)^2(z^2-1.783z+0.8187)^2}{z(z-0.3392)^2(z-1)(z^2-1.783z+0.8187)^2} \tag{21}$$

where the controller is,

$$G_{cz} = \frac{18.288(z-1)(z-0.3392)(z^2-1.783z+0.8187)^2}{z(z-0.3392)^2(z-1)(z^2-1.783z+0.8187)} \tag{22}$$

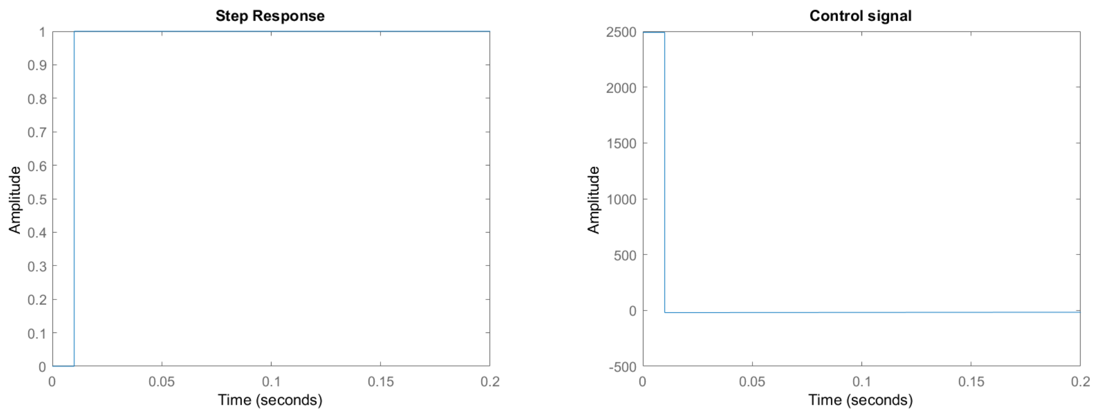
Fig. 6 shows the output of the deadbeat controller and the closed-loop system response to a unit step input. It shows clearly that the steady-state error reaches zero value at the first sampled time.

In addition, the control signal is very high when the sampling time is increased, Fig. 7.



(a) Step response of the closed-loop (b) Control signal

Figure 6: Performance of the deadbeat controller system for $T_s = 1$



(a) Step response of the closed-loop (b) Control signal

Figure 7: Performance of the deadbeat controller system for $T_s = 0.01$

References

[1] Louis C. Westphal. *Handbook of Control Systems Engineering*. Springer Us, Dec. 6, 2012. URL: `https://www.ebook.de/de/product/25178612/louis_c_westphal_handbook_of_control_systems_engineering.html` (cit. on p. 6).