Sample Solutions to Homework Assignment 1

National University of Singapore

Department of Electrical and Computer Engineering

EE3304 Digital Control Systems

Home Assignment I

(Semester II, 2002-2003)

Name: Zhang Xiaoqiang

Matric No.: U00-5668A-11

Email: eng00176@nus.edu.sg

Lecturer: A/P. Ben M. Chen

PART I:

The general form for a PID controller is like the flowing

$$D(s) = K_p + K_d s + \frac{K_i}{s}$$

The plant for this assignment is given as

$$G(s) = \frac{6\times 10^7}{s^2}$$

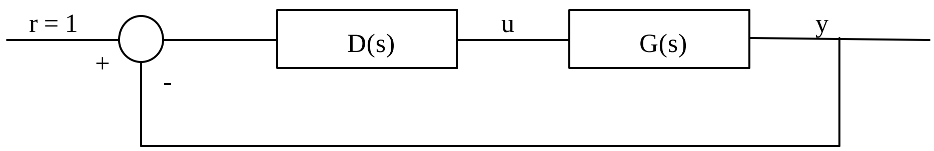


Fig.1: System in continuous time domain

According to the design specification, the settling time is 8 milliseconds, and the maximum overshoot is less than 25%.

Read from the relationship diagram of Mp% vs. ?, we get

$$z \ge 0.4 \Rightarrow z = 0.8$$

$$t_s \cong \frac{4.6}{zw_n}$$

$$\Rightarrow w_n \cong \frac{4.6}{zt_s} = \frac{4.6}{0.8\times 8\times 10^{-3}} = 820$$

Hence the desired transfer function that can meet the design specification is

$$H_{desired}(s) = \frac{820^2}{s^2 + 1312s + 820^2}$$

The closed-loop transfer function for the motor drive system is given by

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)D(s)}{1 + G(s)D(s)}$$

$$H(s) = \frac{6\times 10^7 (K_d s^2 + K_p s + K_i)}{s^3 + 6\times 10^7 (K_d s^2 + K_p s + K_i)}$$

Compare the coefficients on the denominators, and we get

$$\begin{cases} K_i = 0 \\ 6\times10^7 K_d = 1312 \\ 6\times10^7 K_p = 820^2 \end{cases} \Rightarrow \begin{cases} K_i = 0 \\ K_d = 2.18\times10^{-5} \\ K_p = 0.0112 \end{cases}$$

Hence, the closed-loop transfer function is,

$$H(s) = \frac{1312 s^2 + 820^2 s}{s^2 + 1312 s + 820^2}$$

The simulation result though *SIMULINK* is shown as below,

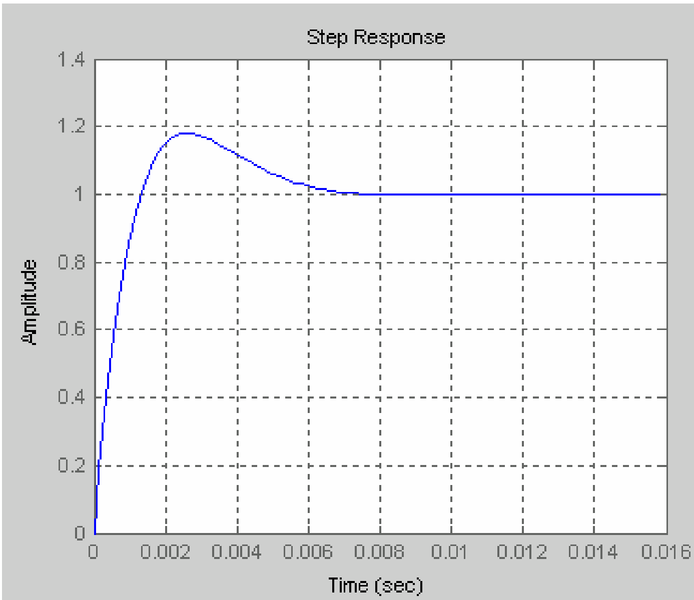


Fig. 2: Step response in the continuous time domain

The amplitude bode plot is,

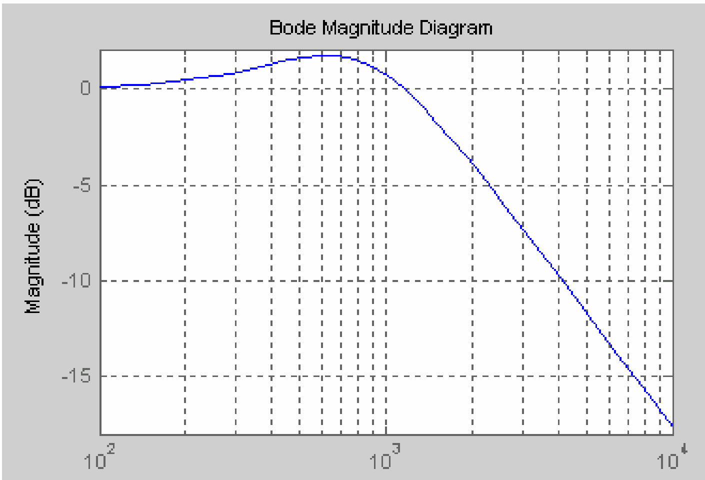


Fig. 3: Bode plot for the closed-loop system

From the bode plot we can get the bandwidth for the frequency is 1790Hz, hence the sufficient sampling frequency is

$$f_{sufficient} = 30 \times BW$$

Discretize the continuous time PD controller with $T=5.59E\text{-}4$ using bilinear transformation method, we get

$$D(z) = D(s)\,|_{s=\frac{2}{T}(\frac{z-1}{z+1})} = K_p + K_d s\,|_{s=\frac{2}{T}(\frac{z-1}{z+1})}$$

Hence,

$$D(z) = \frac{1.358z^2 - 4.693z + 2.335}{z^2 - 1}$$

The simulation result from *SIMULINK* is

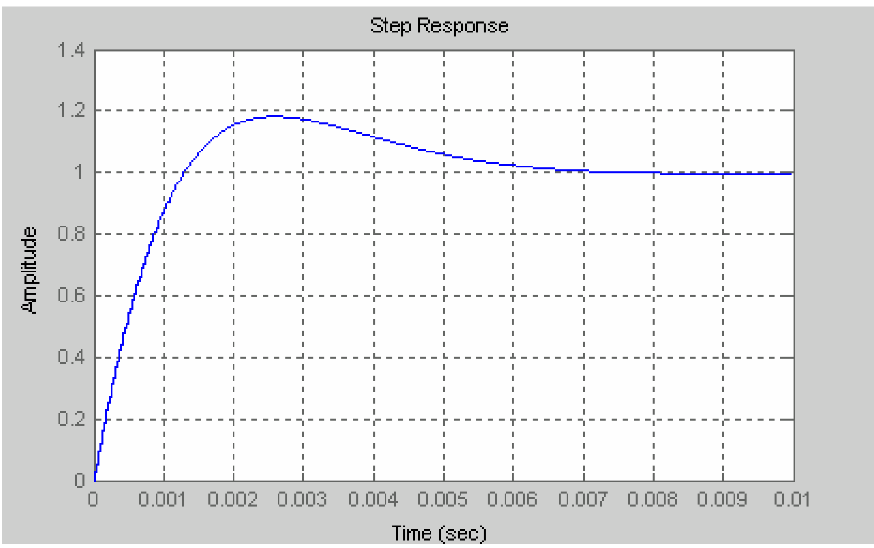


Fig. 4: Step response for the digital controller at $T=5.59E\text{-}4s$

But according to the question, the maximum sampling frequency is $10KHz$.

Hence, the PD controller is

$$D(z) = \frac{0.4485z^2 - 0.8747z + 0.4261}{z^2 - 1}$$

The corresponding simulation result is,

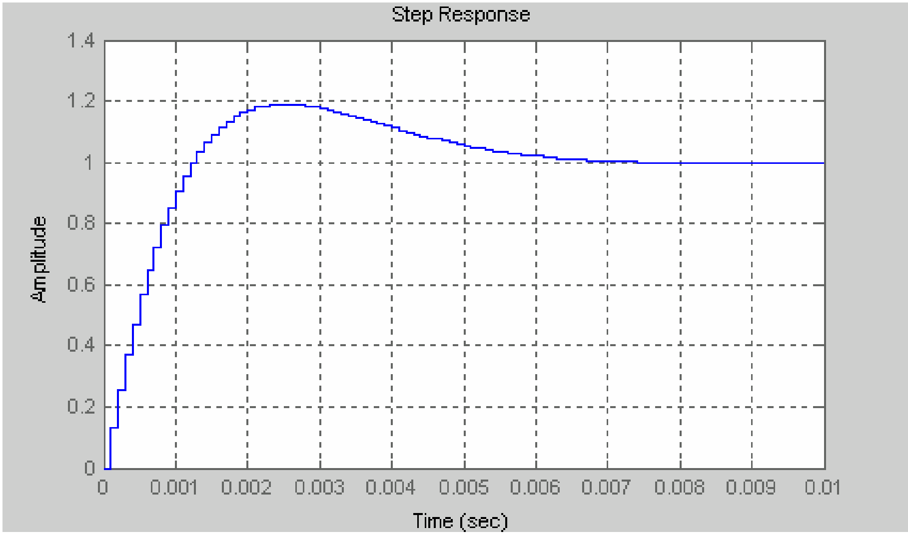


Fig. 5: Step response for the digital controller at *T=0.0001s*

Though the performance is not as good as the previews one, but it still meets the design requirements. For this digital PD controller,

$$\begin{cases} M_p\% \cong 20\% < 25\% \\ t_s = 6ms < 8ms \end{cases}$$

PART II:

Discretize the continuous time plant,

$$G(z) = (1 - Z^{-1}) Z\left\{\frac{G(s)}{s}\right\}$$

Hence, we can get

$$G(z) = \frac{0.3z + 0.3}{z^2 - 2z + 1}$$

From the lecture notes, we know the general form for a PID controller in discrete time domain is,

$$D(z)_{general} = K_p + K_i \frac{z}{z-1} + K_d \frac{z-1}{z}$$

We choose PD controller in this assignment,

$$D(z) = \frac{(K_p + K_d)z - K_d}{z}$$

The system now becomes like

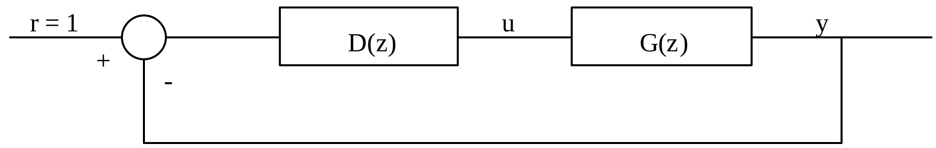


Fig.6: System in discrete time domain

The resulting closed-loop transfer function from r to y is given by

$$H(z) = \frac{D(z)G(z)}{1 + D(z)G(z)}$$

$$\Rightarrow H(z) = \frac{\dfrac{0.3[(K_p + K_d)z^2 + K_p z - K_d]}{z(z-1)^2}}{1 + \dfrac{0.3[(K_p + K_d)z^2 + K_p z - K_d]}{z(z-1)^2}}$$

$$\Rightarrow H(z) = \frac{0.3[(K_p + K_d)z^2 + K_p z - K_d]}{z(z-1)^2 + 0.3[(K_p + K_d)z^2 + K_p z - K_d]}$$

$$\Rightarrow H(z) = \frac{0.3[(K_p + K_d)z^2 + K_p z - K_d]}{z^3 + (0.3K_p + 0.3K_d - 2)z^2 + (0.3K_p + 1)z - 0.3K_d}$$

From PART I, we get the desired $z = 0.8$, and $w_n = 820$ in continuous time domain, which would achieve the design specification.

The poles for the system in continuous time domain is given by,

$$s_p = -zw_n + jw_n\sqrt{1-z^2}$$
$$\Rightarrow s_p = -656 + j492$$

The transformation between s and z is given by

$$z_p = e^{Ts}$$

From which, we can get the poles in the discrete time domain

$$z_p = 0.9354 + j0.0461$$

Hence, the corresponding desired transfer function is

$$H_{desired}(z) = \frac{(1 - z_p)(1 - \bar{z}_p)}{z(z - z_p)(z - \bar{z}_p)}$$

$$\Rightarrow H_{desired}(z) = \frac{0.0063}{z(z^2 - 1.8707z + 0.8770)}$$

Substitute one of the poles of the desired transfer function into the original closed-loop transfer function $H(z)$. They should have the same poles in order to have the similar performance.

$$
\begin{aligned}
& z^3 + (0.3K_p + 0.3K_d - 2)z^2 + (0.3K_p + 1)z - 0.3K_d = 0 \\
& \Rightarrow (0.9354 + j0.0461)^3 + (0.3K_p + 0.3K_d - 2)(0.9354 + j0.0461)^2 + \\
& \quad (0.3K_p + 1)(0.9354 + j0.0461) - 0.3K_d = 0 \\
& \Rightarrow (0.8125 + j0.1209) + (0.3K_p + 0.3K_d - 2)(0.8728 + j0.0862) + \\
& \quad (0.3K_p + 1)(0.9354 + j0.0461) - 0.3K_d = 0 \\
& \Rightarrow (0.8125 + 0.8728(0.3K_p + 0.3K_d - 2) + 0.9354(0.3K_p + 1) - 0.3K_d) + \\
& \quad j(0.1209 + 0.0862(0.3K_p + 0.3K_d - 2) + 0.0461(0.3K_p + 1)) = 0
\end{aligned}
$$

This is the same as the following set of equations:

$$
\begin{cases}
0.8125 + 0.8728(0.3K_p + 0.3K_d - 2) + 0.9354(0.3K_p + 1) - 0.3K_d = 0 \\
0.1209 + 0.0862(0.3K_p + 0.3K_d - 2) + 0.0461(0.3K_p + 1) = 0
\end{cases}
$$

Simplify the set of equations, we can get

$$
\begin{cases}
5425K_p - 382K_d + 23 = 0 \\
397K_p + 259K_d - 54 = 0
\end{cases}
$$

Solve this equation set we can get

$$
\begin{cases}
K_p = 0.0094 \\
K_d = 0.194
\end{cases}
$$

Substitute K_p and K_d in the original transfer function, the denominator has the following polynomial

$$
z^3 - 1.939z^2 + 1.0028z - 0.0528
$$

The three roots for this polynomial is,

$$
\begin{cases}
p_0 = 0.0622 \\
p_1 = 0.9364 + j0.0448 \\
P_2 = 0.9364 - j0.0448
\end{cases}
$$

The two complex roots are nearly the same as those of the desired transfer function. If we take in the consideration of the limited accuracy that occurs in the above computation, the two roots are the same. P_0 is far less than the real part of the other two poles, and it is almost zero, which means the pole has almost no effect on the overall system response. Practically, we replace it with a pole at the location of zero. Hence the PD controller is given as

$$D(z) = \frac{(K_p + K_d)z - K_d}{z}$$

$$\Rightarrow D(z) = \frac{0.2035\,z - 0.194}{z}$$

The system with a digital controller with the digital plant

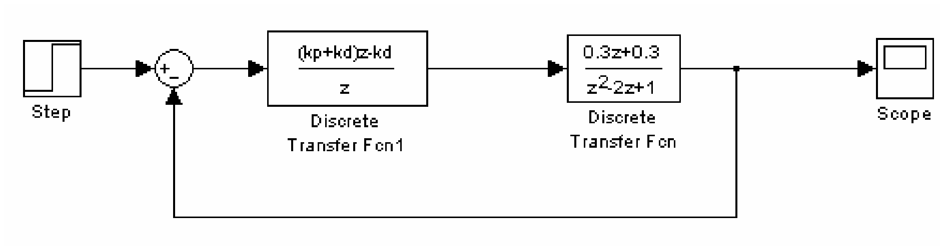


Fig. 7: Discretized plant with a PD digital controller

The simulation result is,

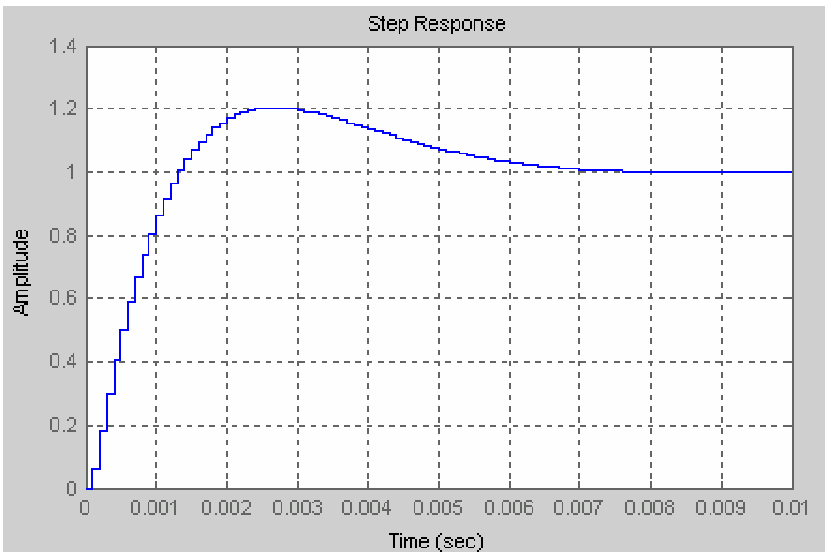


Fig. 8: Simulation result for the discretized plant with a PD digital controller

Since the plant is a operating in continuous time domain, hence the actually system should be like,

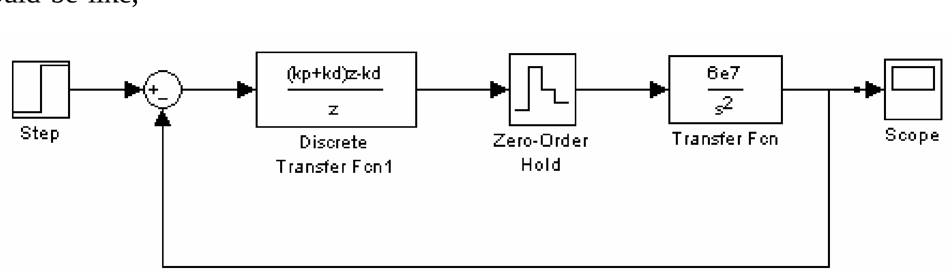
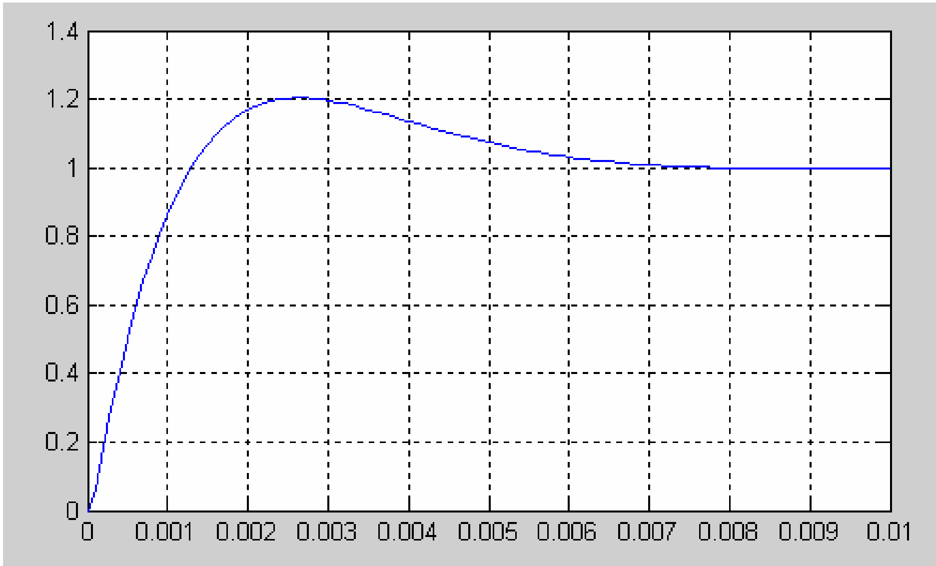


Fig. 7: The actual plant with a PD digital controller

The simulation result is,



According to the result of simulation,

$$\begin{cases} M_p\% \cong 21\% < 25\% \\ t_s = 7ms < 8ms \end{cases}$$

The digital PD controller $D(z) = \dfrac{0.2035z - 0.194}{z}$ can meet the design specification.

Appendix A: M-File for Part I

```
clear all;
close all;
%Define the plant in countinous time domain
num_g = [6E7];
den_g = [1 0 0];
G = tf(num_g, den_g);
%Define the PID controller in countinous time domain
kp = 8.2^2*10^(-3)/6;
ki = 0;
kd = 1.6*8.2/6E5;
num_d = [kd kp ki];
den_d = [1 0];
D = tf( num_d, den_d );
%Define the close-loop transfer function in countinous time domain
FSYS = feedback(D*G, 1);
%Plot the resonse
figure;
step(FSYS);
grid;
figure;
bodemag(FSYS);
grid;
BW = bandwidth(FSYS);
%Convert the countinous time system into discrete time system
D_D = c2d(D, 0.0001, 'tustin');
D_G = c2d(G, 0.0001, 'zoh' );
%D_D = c2d(D, 1/(30*BW), 'tustin');
%D_G = c2d(G, 1/(30*BW), 'zoh' );
D_FSYS = feedback(D_D*D_G, 1);
figure;
step(D_FSYS, 10E-3);
grid;
```

Appendix B: M-File for Part II

```
%M-File
clear all;
close all;
%Define the plant in countinous time domain
num_g = [6E7];
den_g = [1 0 0];
G = tf(num_g, den_g);
D_G = c2d(G, 0.0001, 'zoh' );
%Define the PID controller in countinous time domain
kp = 0.0094;
ki = 0;
kd = 0.1940;
num_d = [(kp+ki+kd) -(kp+2*kd) kd];
den_d = [1 -1 0];
D_D = tf( num_d, den_d, 0.0001 );
%Define the close-loop transfer function in countinous time domain
FSYS = feedback(D_D*D_G, 1);
%Plot the resonse
figure;
step(FSYS, 0.01);
grid;
```