EE5103R Computer Control Systems AY2011/2012

Project: Hard Disk Drive Servo Control

Name:
Phang Swee King
A0033585A

November 9, 2011

Contents

1 Introduction 2

2 Nominal Controller Design via Pole Placement Technique using State Feedback Approach 3

3 Digital Controller at Sampling Frequency of 30 KHz 7

4 Analysis on Sampling Frequency of 10 KHz, 5 KHz and 2 KHz 9

5 Bits Number Requirements 13

6 Conclusion 14

7 Source Codes 15

Chapter 1

Introduction

This project is one of the requirement to fulfill the continuous assessment of the module codename Computer Control Systems (EE5103R) offered by the Department of Electrical and Computer Engineering (ECE) in National University of Singapore (NUS), Singapore.

In this project, we are required to design and investigate control laws on the hard disk drive servo system. It is given that the hard disk drive servo system has the following voice-coil-motor actuator:

$$G(s) = \frac{a}{s^2} \frac{\omega_r^2}{s^2 + 2\zeta_r \omega_r s + \omega_r^2}$$

where $a = 11.3043 \times 10^7$, $\zeta_r = 0.015$ is the damping ratio of the resonance mode, and $\omega_r = 2\pi(7560)$ is the natural frequency of the resonance mode. We are required to perform several tasks with this system:

1. Design a nominal controller by using pole-placement technique in continous-time to have closed-loop damping ratio of 0.8 and the closed-loop bandwidth must be 1200 Hz. Show the step response of the closed-loop system with resonance mode and closed-loop bode plot showing the closed-loop bandwidth;

2. It is required to implement the continuous-time control in part 1 using a digital computer. Find the discrete equivalent of the controller by using any appropriate method that we have learnt from the course and perform simulations of this digital control system for a step response using MATLAB and Simulink. The sampling frequency must be 30 KHz. Perform this simulation using ZOH only;

3. Do the above simulations for the sampling frequencies of 2 KHz, 5 KHz, and 10 KHz. Comment on the results;

4. Compute the number of bits required to realize the controller in part 2 in direct form in a digital computer to keep the error in pole locations to be less than 0.0002.

In this project report, each of the requirements listed above will be presented in detailed in the respective chapter, and the source codes to produce the result will be included in the last chapter.

Chapter 2

Nominal Controller Design via Pole Placement Technique using State Feedback Approach

The nominal plant model of the voice-coil-motor actuator without the resonant mode is a double integrator system

$$G(s) = \frac{11.3043 \times 10^7}{s^2}.$$

By converting it to the state-space representation, we can utilize the MATLAB function `ss`, and obtain the following state-space equation

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned}$$

where $A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 8192 \\ 0 \end{bmatrix}$, and $C = \begin{bmatrix} 0 & 13800 \end{bmatrix}$. Here, x is the state variables of the plant, u is the input to the plant, and y is the output of the plant, so that the pole-placement technique using state feedback can be applied to it.

As observed from the C matrix, only the position of the voice-coil-motor actuator is available for control. Pole-placement technique using state feedback will require all states of the plant to be available to the controller, in our case, 2 states. To estimate the immeasurable state, a full state observer is utilized in this project. Combining the full state observer with state feedback control law $u = -Kx$, we have

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(y - C\hat{x}), \\ u &= -K\hat{x}. \end{aligned}$$

where L is the observer gain, K is the state feedback gain, and $\hat{x}$ is the estimated states of the plant.

In the design of the nominal controller, since the resonant mode of the system was not included, the desired closed-loop characteristic equation is

$$C_a(s) = s^2 + 2(0.8)(2\pi)(1486) + (2\pi)^2(1486)^2,$$

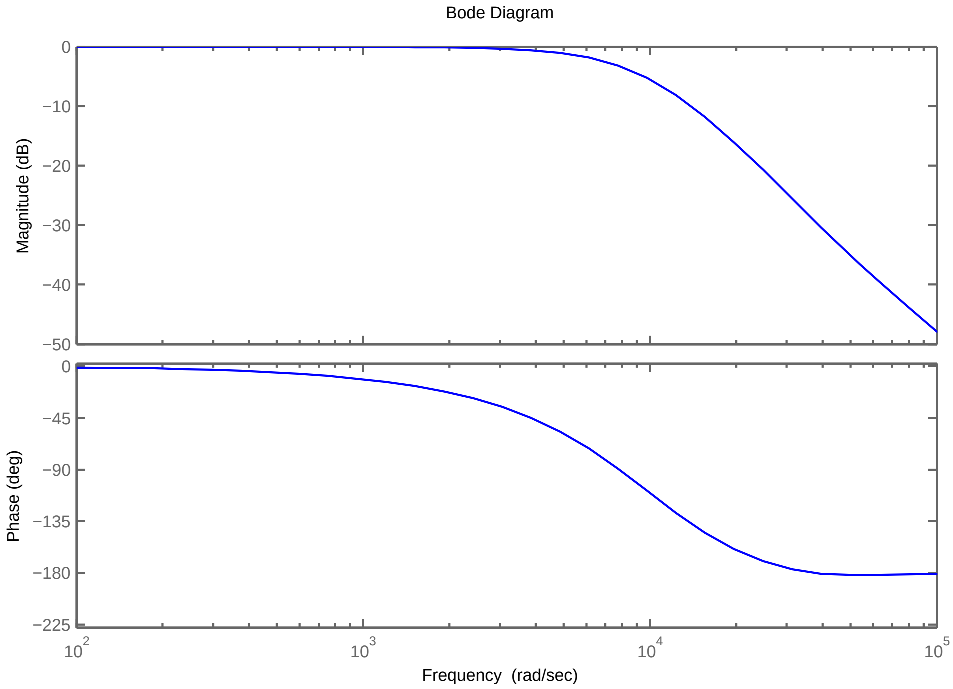

Figure 2.1: Closed-loop bode plot for system without resonant mode

such that it has a damping ratio of 0.8 and the closed-loop bandwidth of 1200 Hz. By the embedded `pole` function in MATLAB, we can obtain the desired pole locations at

$$s = -7469.5 \pm 5602.1j.$$

As for the observer dynamic, a rule of thumb is to position the pole locations at 3 to 5 times faster than the control pole locations. In our case, the observer pole location is chosen as

$$s = -28010,$$

for both the observer poles. By utilizing the `acker` function, we can obtain the desired control gain and observer gain by running the Ackermann's formula, which is

$$K = \begin{bmatrix} 2 & 10642 \end{bmatrix}, \quad L = \begin{bmatrix} 56857 \\ 4 \end{bmatrix}.$$

The bandwidth is given by $\omega_b = 1200.4$ Hz as calculated by MATLAB, which is very close to the requirement.
Fig. 2.1 shows the bode plot of the closed-loop system without the resonant mode. As we can observe, the -3 dB cutoff frequency is at 7540 rad/s which is around 1200 Hz. Fig. 2.2 shows the step response of the system, which characterized the desired closed-loop pole chosen above.

Fig. 2.3 shows the bode plot of the closed-loop system with the resonant mode. Here, we notice that there is a peak (resonant frequency) at around 7080 Hz which is very close to the resonant mode given in the question (7560 Hz). Fortunately, the magnitude of the

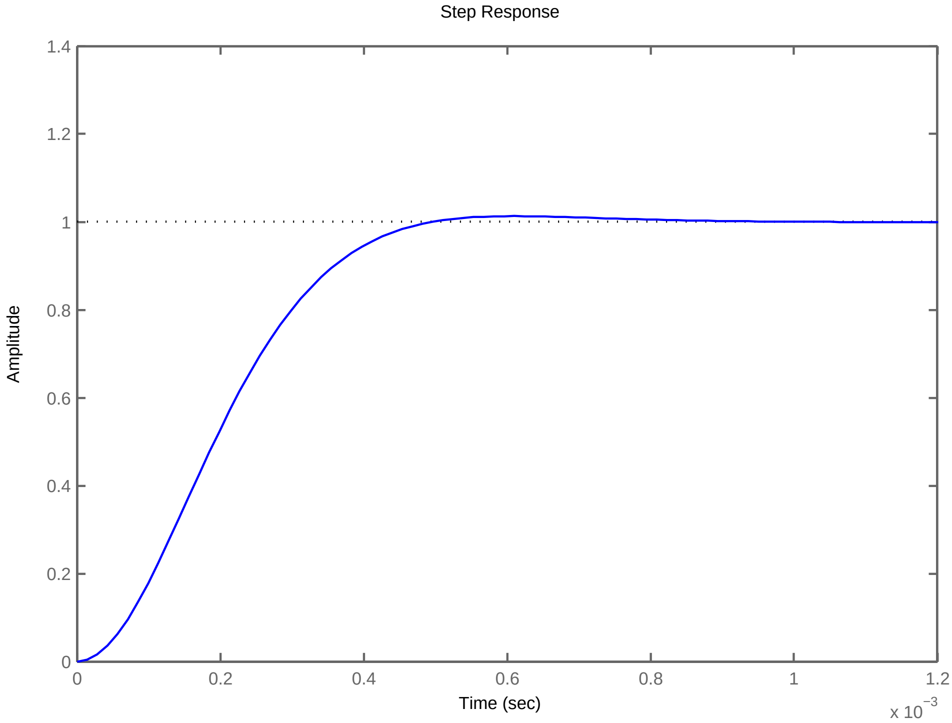

Figure 2.2: Step response for system without resonant mode

resonant frequency is way below 0 dB and thus the system is stable. Fig. 2.4 shows the step response of such system. As we can observe, the resonant mode causes oscillation to the system, and the oscillation frequency can be calculated by taking the time difference between 2 peaks. it is calculated as 7462 Hz which is close to 7560 Hz given. As a result of a more oscillatory step response, the settling time of the response is longer and the overshoot is larger. Nevertheless, the rise time is approximately the same and both step responses are able to achieve zero steady state error as a result of the inclusion of the scaling gain.

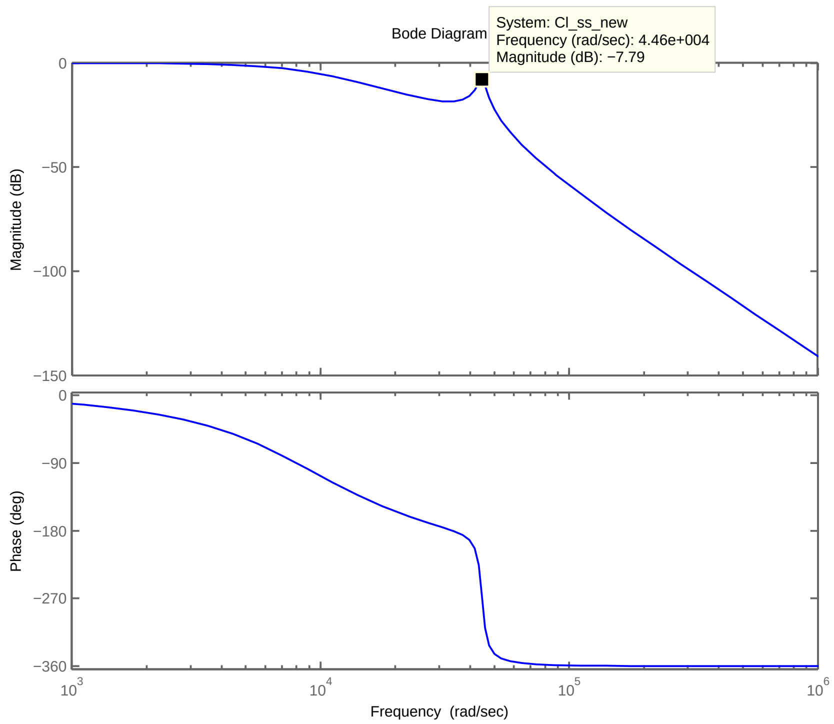

Figure 2.3: Closed-loop bode plot for system with resonant mode

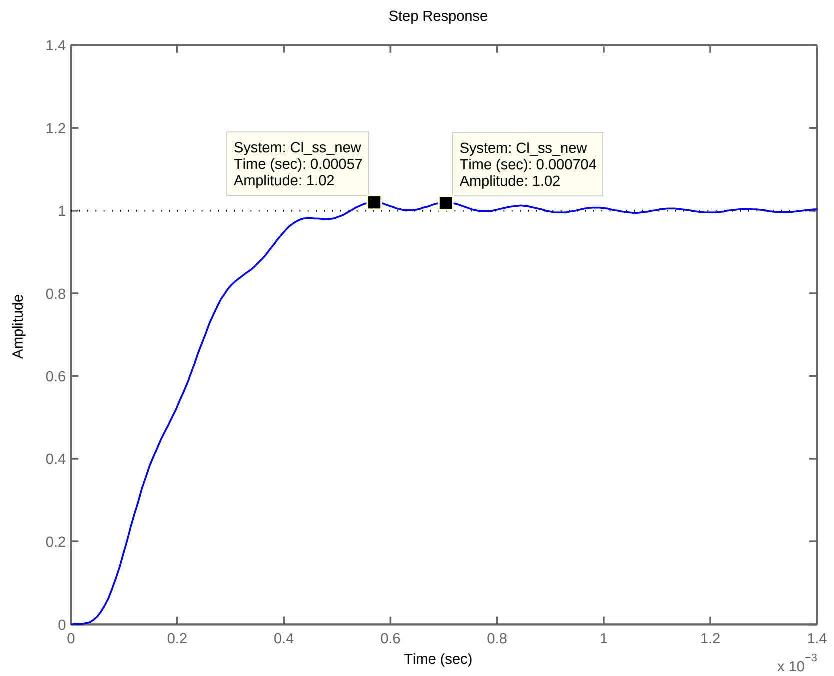

Figure 2.4: Step response for system with resonant mode

Chapter 3

Digital Controller at Sampling Frequency of 30 KHz

The controller designed in the previous chapter will be discretized using the `c2d` function in MATLAB. Here, Tustin approximation method is used for the discretization of the controller. Tustin approximation, also called bilinear transformation, has an advantage that it preserve the stability of the system.

The discrete transfer function of the controller discretized using Tustin approximation at sampling frequency of 30 KHz is

$$C(z) = \frac{-0.9845z^2 - 0.1265z + 0.858}{z^2 - 0.3954z + 0.1099}.$$

The Simulink block diagram for simulation of the closed-loop control configuration is shown in Fig. 7.1. The zero order hold is placed before the plant with resonant mode because the plant model has to be a continuous time model. Output of the Simulink simulation will be directed to MATLAB environment for plotting.

Fig. 3.2 and 3.3 show the step response and the bode plot of the system response with discretized controller. It is seen that with sampling rate of 30 KHz, it is fast enough to preserve the similar response as of the continuous time controller. Notice that in the latter figure, the bode plot of the system is compared to the original bode plot of the continuous system. It seems like that resonant mode is even suppressed and the system has a higher bandwidth.

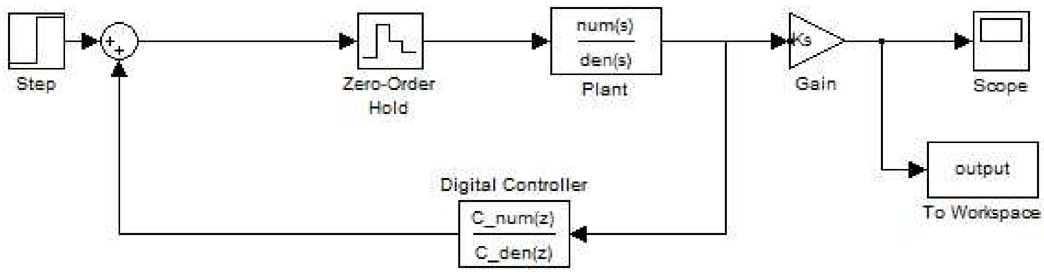

Figure 3.1: Block diagram in Simulink

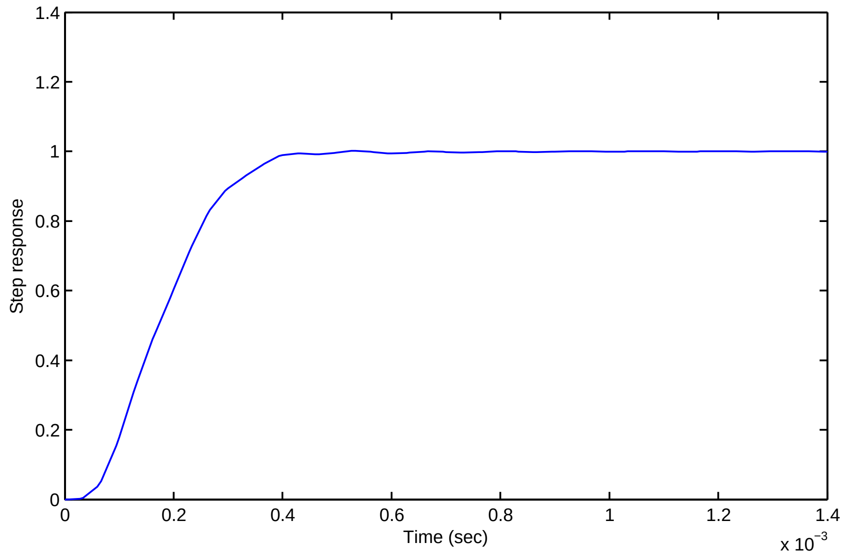

Figure 3.2: Step response of the digital control system

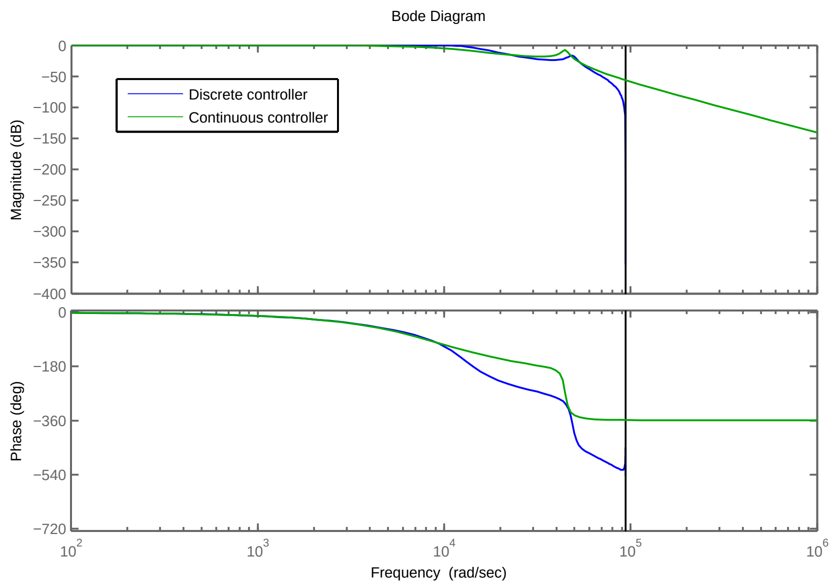

Figure 3.3: Bode plot of digital control system

Chapter 4

Analysis on Sampling Frequency of 10 KHz, 5 KHz and 2 KHz

In this chapter, all the controller are discretized with the Tustin approximation, or in other words, the bilinear transformation. For the sampling frequency of 10 KHz, we have the following discretized controller

$$C(z) = \frac{-1.004z^2 - 0.343z + 0.6612}{z^2 + 0.7419z + 0.1954}.$$

The step response and the bode plot is shown in Fig. 4.1 and 4.2. For a sampling frequency of 10 KHz, the step response becomes very oscillatory with a much larger overshot of about 40 percent as can be seen. It is, however, still a stable system because the sampling frequency is larger than twice the closed-loop bandwidth.

For the sampling frequency of 5 KHz, we can obtain the following transfer function

$$C(z) = \frac{-0.8236z^2 - 0.4805z + 0.343}{z^2 + 1.278z + 0.4364}.$$

The step response and the bode plot is shown in Fig. 4.3 and 4.4. For a sampling frequency of 5 KHz, the step response shows an unstable oscillation with magnitude grows larger with time. It can be concluded that the system can not work well in such low sampling frequency, as it'll destabilized the whole system.

Finally, the system is discretized with 2 KHz sampling frequency. The transfer function is given by

$$C(z) = \frac{-0.5938z^2 - 0.6025z - 0.008716}{z^2 + 1.685z + 0.7174}.$$

The step response and the bode plot is shown in Fig. 4.5 and 4.6. For a sampling frequency of 2 KHz, the step response shows an unstable oscillation with even larger magnitude as to compared to the previous one. It can be concluded that the system can not work well in such low sampling frequency, since the sampling frequency is already lower than twice the system bandwidth, and thus it'll destabilized the whole system.

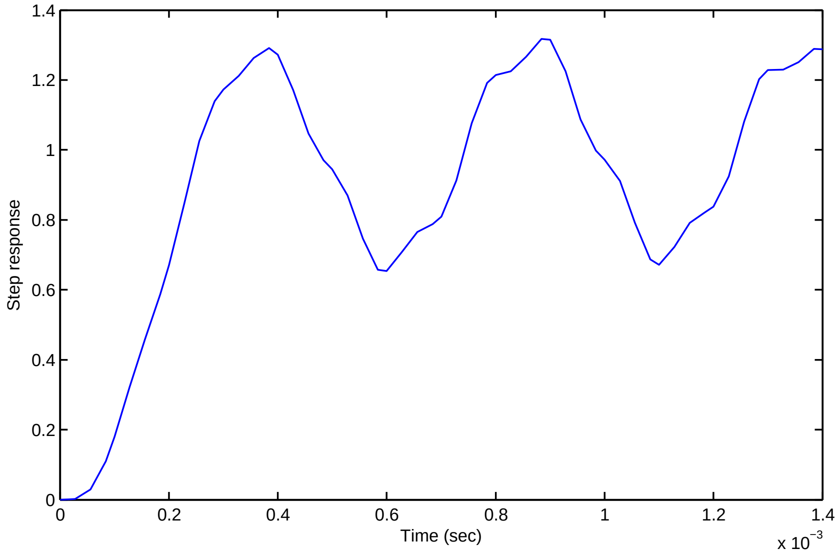

Figure 4.1: Step response with sampling frequency 10 KHz

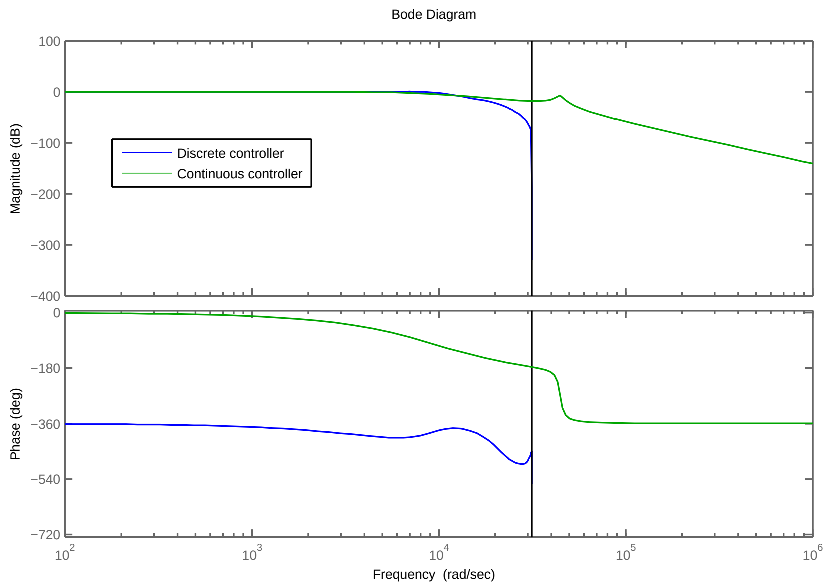

Figure 4.2: Bode plot with sampling frequency 10 KHz

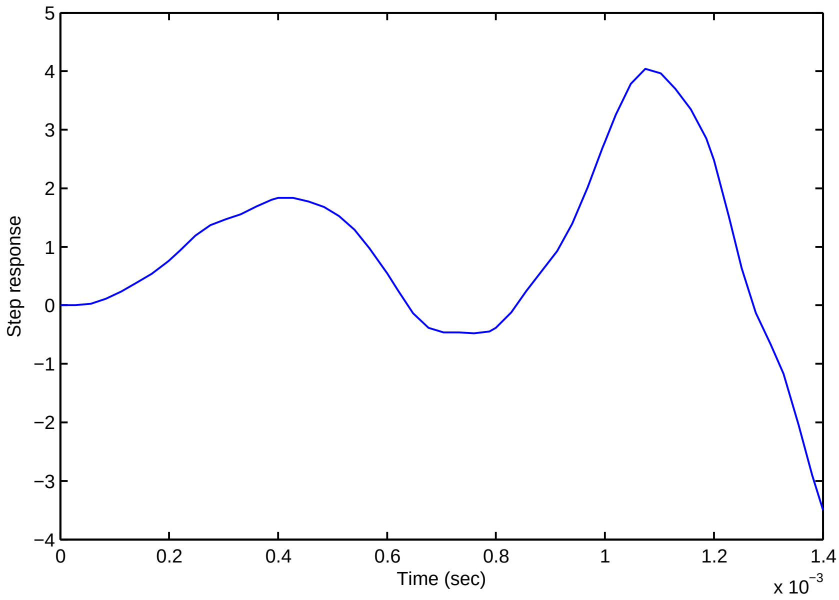

Figure 4.3: Step response with sampling frequency 5 KHz

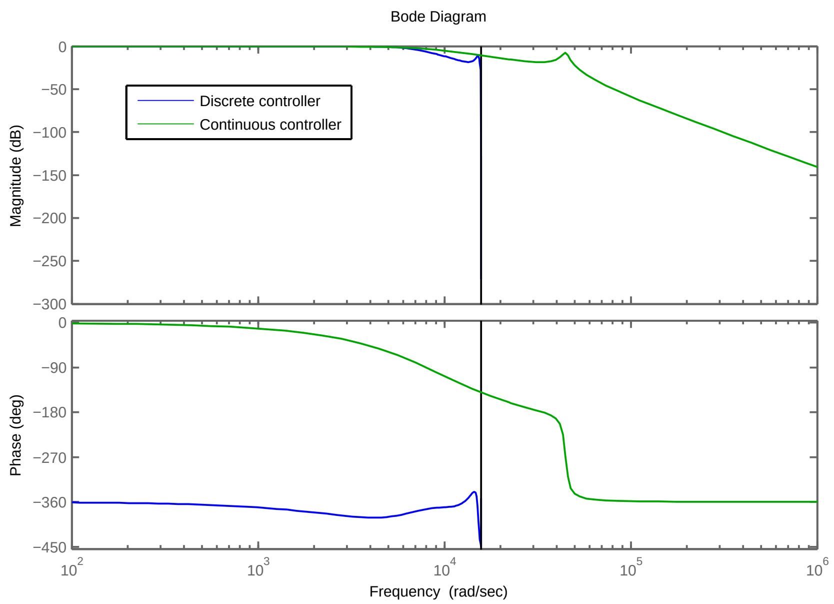

Figure 4.4: Bode plot with sampling frequency 5 KHz

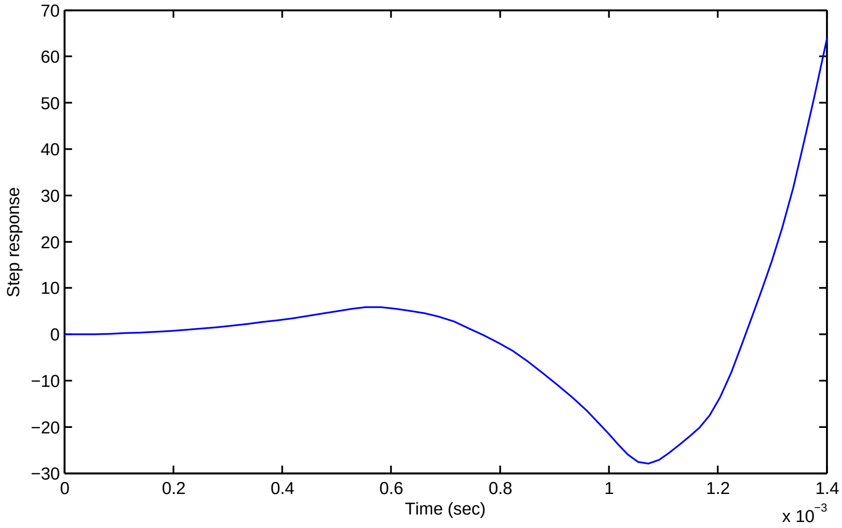

Figure 4.5: Step response with sampling frequency 2 KHz

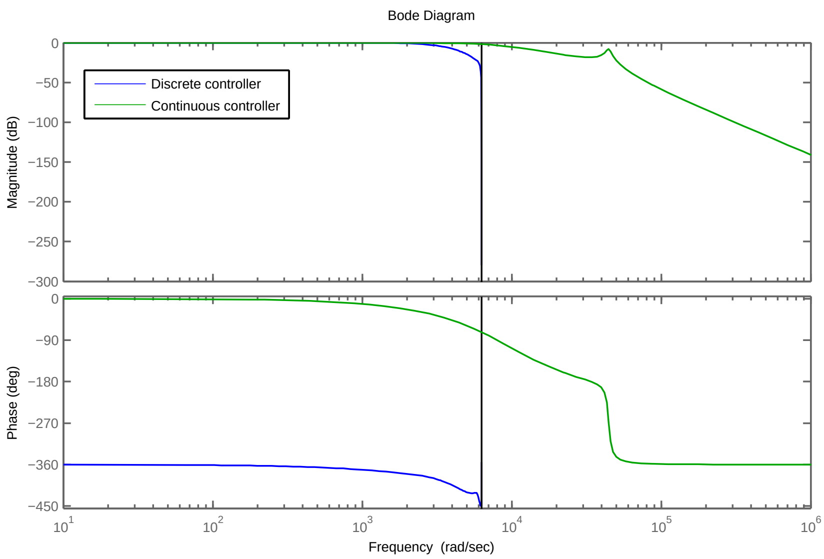

Figure 4.6: Bode plot with sampling frequency 2 KHz

Chapter 5

Bits Number Requirements

In this chapter, the number of bits required to keep the error in pole locations to be less than 0.0002 is calculated.

Taking the sampling frequency to be 30 KHz, we have the pole locations of the controller at

$$z = 0.1977 \pm 0.2661j.$$

Coefficient-pole sensitivity equation is given by

$$\delta p_k = -\frac{p_k^{n-i}}{\prod_{j \neq k}(p_k - p_j)} \delta a_i,$$

where δp_k represents the amount of change in the polo location and δa_i represents the amount of change in the coefficient of the discrete characteristic polynomial.

To compute the number of bits required to realize the controller, the amount of allowed variation in each coefficient of the controller in discrete form, needs to be determined. It is given that the error in pole locations has to be less than 0.0002, thus the amount of allowed variation in each coefficient can be determined using the coefficient-pole sensitivity equation as shown above. Since there are complex poles, the values of δa_i will be complex as well. The results are shown in the table below:

	δa_1	δa_2
$\delta p_1 = 0.0002$	$-0.0002577 - 0.0001915j$	$-0.0001064j$
$\delta p_2 = 0.0002$	$-0.0002577 + 0.0001915j$	$0.0001064j$

From the table above, the minimum amount of allowed variation in a_i is 0.0001064. This amount of variation has to be able to be represented by the discrete system. Therefore, the number of bits required to realize the controller will be

$$\begin{aligned} 2^{-n} &= 0.0001064 \\ n &= 13.19. \end{aligned}$$

As we can see, we must have at least 14 bits (> 13.19) to realize the controller.

Chapter 6

Conclusion

In this project, a hard disk drive servo system with resonant mode is first analyzed, then is controlled by a continuous time controller.

The continuous time controller is then discretized to the digital controller, by using Tustin approximation method. Different sampling frequencies, i.e., 30 KHz, 10 KHz, 5 KHz, and 2 KHz are tested and the controller works in both 30 KHz and 10 KHz sampling frequency only.

Lastly, the number of bits that is required in sampling frequency of 30 KHz, such that the pole location of the controller is as accurate in the margin of 0.0002 is calculated, which is approximately 14 bits.

Chapter 7

Source Codes

The following MATLAB and Simulink files are used in this projects:

1. A0033585A.m
2. A0033585A.mdl

A0033585A.m

```
% This is Computer Control Systems project

%% Part 1: State-feedback
sys1 = tf([11.3043e7],[1 0 0]);
sys1_ss = ss(sys1);

% w = 1486 for CL bandwidth of 1200 Hz (desired system)
w = 1486;
sys_ref = tf([4*pi*pi*w*w],[1 2*0.8*2*pi*w 4*pi*pi*w*w]);

% Desired closed-loop poles
P = pole(sys_ref);

% State feedback controller K
K = acker(sys1_ss.a, sys1_ss.b, P);

% Observer poles
Po = -abs(P)*3;

% Observer gain L
L = acker(sys1_ss.a', sys1_ss.c', Po)';

% Overall controller
Ac = sys1_ss.a - sys1_ss.b * K - L * sys1_ss.c;
Bc = L;
Cc = -K;
Dc = 0;
C_ss = ss(Ac, Bc, Cc, Dc);

% Closed loop system
```

```
Cl_ss = feedback(sys1_ss, C_ss, 1);

% Multiply by DC gain for DC 1
Ks = 1/dcgain(Cl_ss);
Cl_ss = Ks*Cl_ss;

% Check pole location
pole(Cl_ss);
% Check bandwidth
bandwidth(Cl_ss)/(2*pi);

% Plot bode and step
figure(1);
bode(Cl_ss);
figure(2);
step(Cl_ss);

% Combine with resonant mode
sys2 = tf((2*pi*7560)^2,[1 2*0.015*2*pi*7560 (2*pi*7560)^2]);
sys2_ss = ss(sys2);

%New closed loop:
Cl_ss_new = feedback(sys1_ss*sys2_ss, C_ss, 1);
Cl_ss_new = Ks*Cl_ss_new;
figure(3);
bode(Cl_ss_new);
figure(4);
step(Cl_ss_new);


%% Question 2 and 3: digital controller
% Sampling time, Ts (varied for question 3)
Ts = 1/30000;

% Sampling with tustin method
C_tf = tf(C_ss);
Cd_tf = c2d(C_tf, Ts, 'tustin')
[C_num, C_den] = tfdata(Cd_tf, 'v');

% Plant sampled with zoh
Gnom_d = c2d(sys1,Ts,'zoh');
Gres_d = c2d(sys2,Ts,'zoh');

Cld_tf = feedback(Gnom_d*Gres_d, Cd_tf, 1);
Cld_tf = Ks*Cld_tf;

% Plot step and bode of both cont and digi plant
sim('a0033585a.mdl')
figure(5)
y = output.signals.values;
t = output.time;
plot(t,y)
xlabel('Time (sec)')
ylabel('Step response')

figure(6)
bode(Cld_tf, Cl_ss_new)
legend('Discrete controller','Continuous controller');
```

```
%% question 4, number of bits
P = pole(Cd_tf)
```

A0033585A.mdl

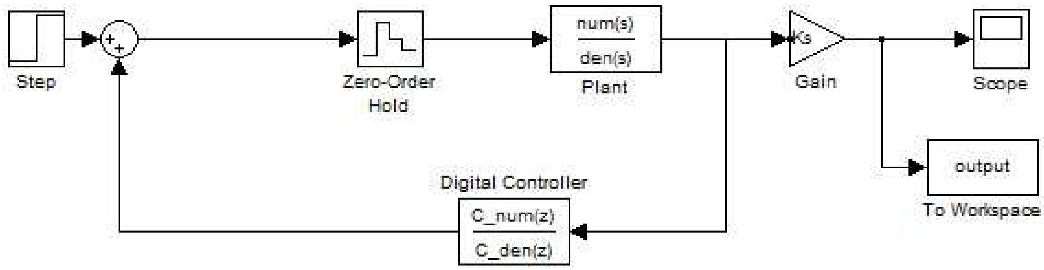

Figure 7.1: Block diagram in Simulink