• 3 •

Introductory Digital Control

A Perspective on Introductory Digital Control

The continuous controllers you have studied so far are built using analog electronics such as resistors, capacitors, and operational amplifiers. However, most control systems today use digital computers (usually microprocessors or microcontrollers) with the necessary input/output hardware to implement the controllers. The intent of this chapter is to show the very basic ideas of designing control laws that will be implemented in a digital computer. Unlike analog electronics, digital computers cannot integrate. Therefore, in order to solve a differential equation in a computer, the equation must be approximated by reducing it to an algebraic equation involving sums and products only. These approximation techniques are often referred to as **numerical integration**. This chapter shows a simple way to make these approximations as an introduction to digital control. Later chapters expand on various improvements to these approximations, show how to analyze them, and show that digital compensation may also be carried out directly without resorting to these approximations. In the final analysis, we will see that direct digital design provides the designer with the most accurate method and the most flexibility in selection of the sample rate.

From the material in this chapter, you should be able to design and implement a digital control system. The system would be expected to give adequate performance if the sample rate is at least 30 times faster than the bandwidth of the system.

Chapter Overview

In Section 3.1, you will learn how to approximate a continuous $D(s)$ with a set of difference equations, a design method sometimes referred to as **emulation**. Section 3.1 is sufficient to enable you to approximate a continuous feedback controller in a digital control system. Section 3.2 shows the basic effect of sampling on the performance of the system and a simple way to analyze that effect. Section 3.3 shows how to convert a continuous PID control law to the digital form.

3.1 Digitization

Figure 3.1(a) shows the topology of the typical continuous system. The computation of the error signal, e, and the dynamic compensation, $D(s)$, can all be accomplished in a digital computer as shown in Fig. 3.1(b). The fundamental differences between the two implementations are that the digital system operates on *samples* of the sensed plant output rather than on the continuous signal and that the dynamics represented by $D(s)$ are implemented by algebraic recursive equations called **difference equations.**

We consider first the action of the analog-to-digital (A/D) converter on a signal. This device acts on a physical variable, most commonly an electrical voltage, and converts it into a binary number that usually consists of 10 or 12 bits. A binary number with 10 bits can take on $2^{10} = 1024$ values; therefore, an A/D converter with 10 bits has a resolution of 0.1%. The conversion from the analog signal $y(t)$ occurs repetitively at instants of time that are T seconds

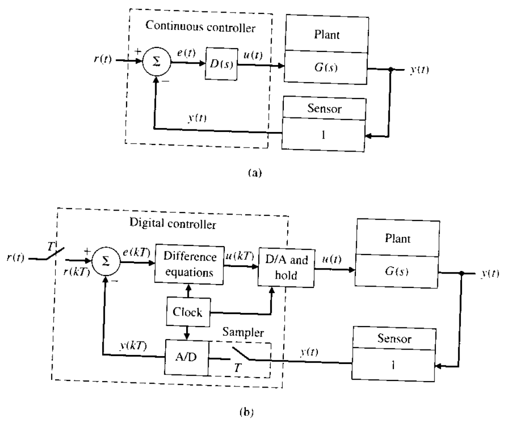


Figure 3.1
Basic control-system block diagrams:
(a) continuous system,
(b) with a digital computer

sample period
sample rate

apart. T is called the **sample period** and $1/T$ is the **sample rate** in cycles per second or Hz (also sometimes given in radians/second or $2\pi/T$). The sampled signal is $y(kT)$ where k can take on any integer value. It is often written simply as $y(k)$. We call this type of variable a **discrete signal** to distinguish it from a continuous variable like $y(t)$, which changes continuously in time. We make the assumption here that the sample period is fixed; however, it may vary depending on the implementation as discussed in Section 1.1.

ZOH

There also may be a sampler and A/D converter for the input command, $r(t)$, producing the discrete $r(kT)$ from which the sensed $y(kT)$ would be subtracted to arrive at the discrete error signal, $e(kT)$. The differential equation of the continuous compensation is approximated by a difference equation which is the discrete approximation to the differential equation and can be made to duplicate the dynamic behavior of a $D(s)$ if the sample period is short enough. The result of the difference equation is a discrete $u(kT)$ at each sample instant. This signal is converted to a continuous $u(t)$ by the D/A and hold. The D/A converts the binary number to an analog voltage, and a **zero-order hold** (ZOH) maintains that same voltage throughout the sample period. The resulting $u(t)$ is then applied to the actuator in precisely the same manner as the continuous implementation.

Euler's method

One particularly simple way to make a digital computer approximate the real time solution of differential equations is to use Euler's method. It follows from the definition of a derivative that

$$\dot{x} = \lim_{\delta t \to 0} \frac{\delta x}{\delta t} \tag{3.1}$$

where δx is the change in x over a time interval δt. Even if δt is not quite equal to zero, this relationship will be approximately true, and

$$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T} \tag{3.2}$$

where

$T = t_{k+1} - t_k$ (the sample interval in seconds),

$t_k = kT$ (for a constant sample interval),

k is an integer,

$x(k)$ is the value of x at t_k, and

$x(k+1)$ is the value of x at t_{k+1}.

difference equations

This approximation[1] can be used in place of all derivatives that appear in the controller differential equations to arrive at a set of equations that can be solved by a digital computer. These equations are called **difference equations** and are solved repetitively with time steps of length T. For systems having bandwidths

1 This particular version is called the **forward rectangular** rule. See Problem 3.2 for the **backward rectangular** version.

of a few Hertz, sample rates are often on the order of 100 Hz, so that sample periods are on the order of 10 msec and errors from the approximation can be quite small.

◆ **Example 3.1** *Difference Equations Using Euler's Method*

Using Euler's method, find the difference equations to be programmed into the control computer in Fig. 3.1(b) for the case where the $D(s)$ in Fig. 3.1(a) is

$$D(s) = \frac{U(s)}{E(s)} = K_o \frac{s+a}{s+b}. \tag{3.3}$$

Solution. First find the differential equation that corresponds to $D(s)$. After cross multiplying Eq. (3.3) to obtain

$$(s+b)U(s) = K_o(s+a)E(s),$$

we can see by inspection that the corresponding differential equation is

$$\dot{u} + bu = K_o(\dot{e} + ae). \tag{3.4}$$

Using Euler's method to approximate Eq. (3.4) according to Eq. (3.2), we get the approximating difference equation

$$\frac{u(k+1) - u(k)}{T} + bu(k) = K_o\left[\frac{e(k+1) - e(k)}{T} + ae(k)\right]. \tag{3.5}$$

Rearranging Eq. (3.5) puts the difference equation in the desired form

$$u(k+1) = u(k) + T\left[-bu(k) + K_o\left(\frac{e(k+1) - e(k)}{T} + ae(k)\right)\right]. \tag{3.6}$$

Equation (3.6) shows how to compute the new value of the control, $u(k+1)$, given the past value of the control, $u(k)$, and the new and past values of the error signal, $e(k+1)$ and $e(k)$. For computational efficiency, it is convenient to re-arrange Eq. (3.6) to

$$u(k+1) = (1 - bT)u(k) + K_o(aT - 1)e(k) + K_o e(k+1). \tag{3.7}$$

◆

In principle, the difference equation is evaluated initially with $k = 0$, then $k = 1, 2, 3, \ldots$. However, there is usually no requirement that values for all times be saved in memory. Therefore, the computer need only have variables defined for the current and past values for this first-order difference equation. The instructions to the computer to implement the feedback loop in Fig. 3.1(b) with the difference equation from Eq. (3.7) would call for a continual looping through the code in Table 3.1. Note in the table that the calculations have been arranged so as to minimize the computations required between the reading of the A/D and the writing to the D/A, thus keeping the computation delay to a minimum.

Table 3.1 Real Time Controller Implementation

$x = 0$ (initialization of past values for first loop through)
Define constants:
$\alpha_1 = 1 - bT$
$\alpha_2 = K_o(aT - 1)$
READ A/D to obtain y and r
$e = r - y$
$u = x + K_o e$
OUTPUT u to D/A and ZOH
now compute x for the next loop through
$x = \alpha_1 u + \alpha_2 e$
go back to READ when T seconds have elapsed since last READ

The sample rate required depends on the closed-loop bandwidth of the system. Generally, sample rates should be faster than 30 times the bandwidth in order to assure that the digital controller can be made to closely match the performance of the continuous controller. Discrete design methods described in later chapters will show how to achieve this performance and the consequences of sampling even slower if that is required for the computer being used. However, when using the techniques presented in this chapter, a good match to the continuous controller is obtained when the sample rate is greater than approximately 30 times the bandwidth.

◆ **Example 3.2** *Lead Compensation Using a Digital Computer*

Find digital controllers to implement the lead compensation

$$D(s) = 70\frac{s+2}{s+10} \tag{3.8}$$

for the plant

$$G(s) = \frac{1}{s(s+1)}$$

using sample rates of 20 Hz and 40 Hz. Implement the control equations on an experimental laboratory facility like that depicted in Fig. 3.1, that is, one that includes a microprocessor for the control equations, a ZOH, and analog electronics for the plant. Compute the theoretical step response of the continuous system and compare that with the experimentally determined step response of the digitally controlled system.

Solution. Comparing the compensation transfer function in Eq. (3.8) with Eq. (3.3) shows that the values of the parameters in Eq. (3.6) are $a = 2$, $b = 10$, and $K_o = 70$. For a sample rate of 20 Hz, $T = 0.05$ sec and Eq. (3.6) can be simplified to

$$u(k+1) = 0.5u(k) + 70[e(k+1) - 0.9e(k)].$$

For a sample rate of 40 Hz, $T = 0.025$ sec and Eq. (3.6) simplifies to

$$u(k+1) = 0.75u(k) + 70[e(k+1) - 0.95e(k)].$$

The statements in MATLAB[2] to compute the continuous step response is

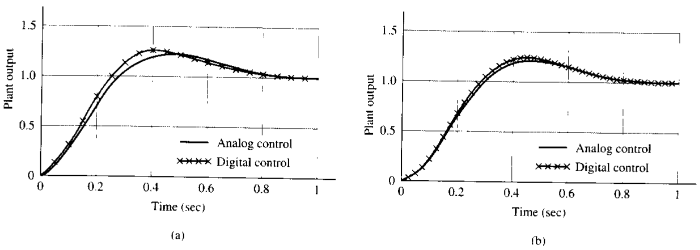
```
numD = 70*[1   2],   denD = [1   10]
numG = 1,   denG = [1   1   0]
sys1 = tf(numD,denD)*tf(numG,denG)
sysCL = feedback(sys1,1)
step(sysCL).
```

Figure 3.2 shows the step response of the two digital controllers compared to the continuous step response. Note that the 40 Hz sample rate (about 30 × bandwidth) behaves essentially like the continuous case, whereas the 20 Hz sample rate (about 15 × bandwidth) has a detectable increased overshoot signifying some degradation in the damping. The damping would degrade further if the sample rate were made any slower.

The MATLAB file that created Fig. 3.2 (fig32.m) computed the digital responses as well as the continuous response. You will learn how to compute the response of a digital system in Chapter 4.

◆

Figure 3.2
Continuous and digital step response using Euler's method for discretization: (a) 20 Hz sample rate, (b) 40 Hz sample rate



2 Assumes the use of MATLAB v5 and Control System Toolbox v4. For prior versions, see Appendix F.

In Chapter 6, you will see that there are several ways to approximate a continuous transfer function, each with different merits, and most with better qualities than the Euler method presented here. In fact, MATLAB provides a function (c2d.m) that computes these approximations. However, before those methods can be examined, it will be necessary to understand discrete transfer functions, a topic covered in Chapter 4.

3.2 Effect of Sampling

It is worthy to note that *the single most important* impact of implementing a control system digitally is the delay associated with the hold. A delay in any feedback system degrades the stability and damping of the system. Because each value of $u(kT)$ in Fig. 3.1(b) is held constant until the next value is available from the computer, the continuous value of $u(t)$ consists of steps (see Fig. 3.3) that, on the average, lag $u(kT)$ by $T/2$, as shown by the dashed line in the figure. By incorporating a continuous approximation of this $T/2$ delay in a continuous analysis of the system, an assessment can be made of the effect of the delay in the digitally controlled system. The delay can be approximated by the method of Padé. The simplest first-order approximation is

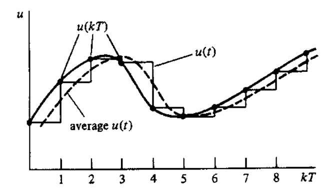
$$G_h(s) = \frac{2/T}{s + 2/T}. \tag{3.9}$$

Figure 3.4 compares the responses from Fig. 3.2 with a continuous analysis that includes a delay approximation according to Eq. (3.9).

This linear approximation of the sampling delay (Eq. (3.9)) could also be used to determine the effect of a particular sample rate on the roots of a system via linear analysis, perhaps a locus of roots vs. T. Alternatively, the effect of a delay can be analyzed using frequency response techniques because a time delay of $T/2$ translates into a phase decrease of

$$\delta\phi = -\frac{\omega T}{2}. \tag{3.10}$$

Figure 3.3
The delay due to the hold operation



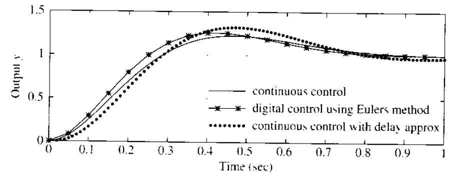


Figure 3.4 Continuous and digital step response at 20 Hz sample rate showing results with a $T/2$ delay approximation

Thus, we see that the loss of phase margin due to sampling can be estimated by invoking Eq. (3.10) with ω equal to the frequency where the magnitude equals one, that is, the "gain crossover frequency."

◆ **Example 3.3** *Approximate Analysis of the Effect of Sampling*

For the system in Example 3.2, determine the decrease in damping that would result from sampling at 10 Hz. Use both linear analysis and the frequency response method. Compare the time response of the continuous system with the discrete implementation to validate the analysis.

Solution. The damping of the system in Example 3.2 can be obtained from the MATLAB statement

```
damp(sysCL)
```

where sysCL is that computed in Example 3.2. The result is $\zeta = .56$.

The damping of the system with the simple delay approximation added (Eq. (3.9)) is obtained from

```
T = 1/10
numDL = 2/T,    denDL = [1   2/T]
sys2 = tf(numDL,denDL)*sys1
sysCL = feedback(sys2,1)
damp(sysCL)
```

where sys1 is that computed in Example 3.2. The result of this calculation is $\zeta = .33$.

The frequency response of the continuous system is shown by the solid line in Fig. 3.5 and shows that the crossover frequency is about 6 rad/sec and the PM is about 50°. The line of small circles shows the phase corrected by Eq. (3.10) and, therefore, that the PM decreases to about 30°. For more precision, the use of margin.m in MATLAB shows that the continuous system has a PM of 49.5° at a crossover frequency of 6.17 rad/sec. Equation (3.10) then indicates that the correction due to sampling should be 17.7°, thus the PM of the digital system would be

31.8°. Since the PM is approximately $100 \times \zeta$, this analysis shows that the ζ decreases from approximately 0.5 for the continuous system to 0.32 for the digital system.

Both analysis methods indicate a similar reduction in the damping of the system. One should, therefore, expect that the overshoot of the step response should increase. For the case with no zeros, Fig. 2.7 indicates that this decrease in ζ should result in the step response overshoot, M_p, going from 16% to 35% for a 2nd-order system with no zeros. The actual step responses in Fig. 3.6 have about 20% overshoot for the continuous system and about 30% for the digital case. So, we see that the approximate analysis was somewhat conservative in the prediction of the decreased damping and increased overshoot in the digital case. The trend that decreasing sample rate causes decreasing damping and stability will be analyzed more completely throughout the book. ◆

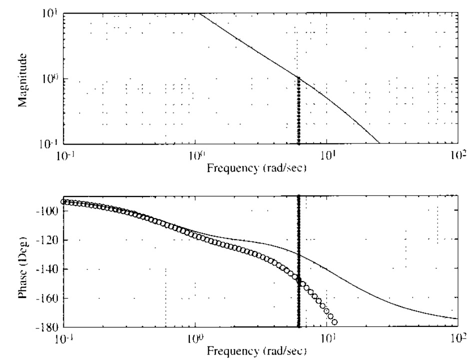


Figure 3.5 Frequency response for Example 3.3

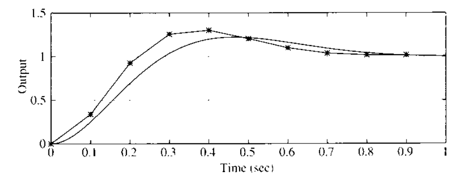


Figure 3.6 Continuous and digital responses for Example 3.3 (at 10 Hz sample rate)

3.3 PID Control

The notion of proportional, integral, and derivative (PID) control is reviewed in Section 2.2.3. Reviewing again briefly, the three terms are proportional control

$$u(t) = Ke(t), \tag{3.11}$$

integral control

$$u(t) = \frac{K}{T_I}\int_0^t e(\eta)d\eta, \tag{3.12}$$

and derivative control

$$u(t) = KT_D\dot{e}(t), \tag{3.13}$$

where K is called the proportional gain, T_I the integral time, and T_D the derivative time. These three constants define the control.

The approximations of these individual control terms to an algebraic equation that can be implemented in a digital computer are proportional control

$$u(k) = Ke(k). \tag{3.14}$$

integral control

$$u(k) = u(k-1) + \frac{K}{T_I}Te(k), \tag{3.15}$$

and derivative control

$$u(k) = \frac{KT_D}{T}[e(k) - e(k-1)]. \tag{3.16}$$

Equation (3.11) is already algebraic, therefore Eq. (3.14) follows directly while Eqs. (3.15) and (3.16) result from an application of Euler's method (Eq. (3.2)) to Eqs. (3.12) and (3.13). However, normally these terms are used together and, in this case, the combination needs to be done carefully. The combined continuous transfer function (Eq. 2.24) is

$$D(s) = \frac{u(s)}{e(s)} = K(1 + \frac{1}{T_I s} + T_D s).$$

Therefore, the differential equation relating $u(t)$ and $e(t)$ is

$$\dot{u} = K(\dot{e} + \frac{1}{T_I}e + T_D\ddot{e})$$

and the use of Euler's method (twice for $\ddot{e}$) results in

$$u(k) = u(k-1) + K\left[\left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right)e(k) - \left(1 + 2\frac{T_D}{T}\right)e(k-1) + \frac{T_D}{T}e(k-2)\right]. \tag{3.17}$$

◆ **Example 3.4** *Transforming a Continuous PID to a Digital Computer*

A micro-servo motor has a transfer function from the input applied voltage to the output speed (rad/sec),

$$G(s) = \frac{360000}{(s+60)(s+600)}. \tag{3.18}$$

It has been determined that PID control with $K = 5$, $T_D = 0.0008$ sec, and $T_I = 0.003$ sec gives satisfactory performance for the continuous case. Pick an appropriate sample rate, determine the corresponding digital control law, and implement on a digital system. Compare the digital step response with the calculated response of a continuous system. Also, separately investigate the effect of a higher sample rate and re-tuning the PID parameters on the ability of the digital system to match the continuous response.

Solution. The sample rate needs to be selected first. But before we can do that, we need to know how fast the system is or what its bandwidth is. The solid line in Fig. 3.7 shows the step response of the continuous system and indicates that the rise time is about 1 msec. Based on Eq. (2.16), this suggests that $\omega_n \cong 1800$ rad/sec, and so the bandwidth would be on the order of 2000 rad/sec or 320 Hz. Therefore, the sample rate would be about 3.2 kHz if 10 times bandwidth. So let's pick $T = 0.3$ msec. Use of Eq. (3.17) results in the difference equation

$$u(k) = u(k-1) + 5[3.7667e(k) - 6.3333e(k-1) + 2.6667e(k-2)]$$

which, when implemented in the digital computer results in the line with stars in Fig. 3.7. This implementation shows a considerably increased overshoot over the continuous case. The line with circles in the figure shows the improved performance obtained by increasing the sample rate to 10 kHz; i.e., a sample rate about 30 times bandwidth, while using the same PID parameters as before. It shows that the digital performance has improved to be essentially the same as the continuous case.

Increasing the sample rate, however, will increase the cost of the computer and the A/D converter; therefore, there will be a cost benefit by improving the performance while maintaining the 3.2 kHz sample rate. A look at Fig. 3.7 shows that the digital response ($T = 0.3$ msec) has a faster rise time and less damping than the continuous case. This suggests that the proportional gain, K, should be reduced to slow the system down and the derivative time, T_D, should be increased to increase the damping. Some trial and error, keeping these ideas in mind, produces the results in Fig. 3.8. The revised PID parameters that produced these results are $K = 3.2$ and $T_D = 0.0011$ sec. The integral reset time, T_I, was left unchanged. ◆

This example once again showed the characteristics of a digital control system. The damping was degraded an increasing amount as the sample rate was reduced. Furthermore, it was possible to restore the damping with suitable adjustments to the control.

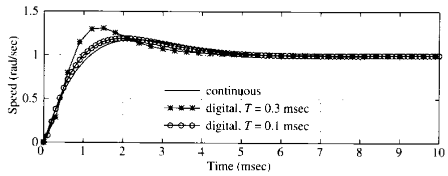


Figure 3.7 Step response of a micro-motor, Example 3.4, same PID parameters

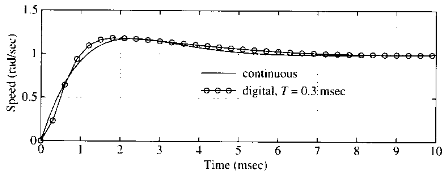


Figure 3.8 Effect of PID tuning on the digital response, Example 3.4

3.4 Summary

- Digitization methods allow the designer to convert a continuous compensation, $D(s)$, into a set of difference equations that can be programmed directly into a control computer.
- Euler's method can be used for the digitization

$$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T}. \tag{3.2}$$

- As long as the sample rate is on the order of 30 × bandwidth or faster, the digitally controlled system will behave close to its continuous counterpart and the continuous analysis that has been the subject of your continuous control systems study will suffice.
- For sample rates on the order of 10 to 30 times the bandwidth, a first order analysis can be carried out by introducing a delay of $T/2$ in the continuous analysis to see how well the digital implementation matches the continuous analysis. A zero-pole approximation for this delay is

$$G_h(s) = \frac{2/T}{s + 2/T}. \tag{3.9}$$

The delay can be analyzed more accurately using frequency response where the phase from the continuous analysis should be decreased by

$$\delta\phi = \frac{\omega T}{2}. \tag{3.10}$$

- A continuous PID control law whose transfer function is

$$D(s) = \frac{u(s)}{e(s)} = K(1 + \frac{1}{T_I s} + T_D s)$$

can be implemented digitally using Eq. (3.17)

$$u(k) = u(k-1) + K\left[\left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right)e(k) - \left(1 + 2\frac{T_D}{T}\right)e(k-1) + \frac{T_D}{T}e(k-2)\right].$$

The digital control system will behave reasonably close to the continuous system providing the sample rate is faster than 30 times the bandwidth.

- In order to analyze the system accurately for any sample rate, but especially for sample rates below about 30 times bandwidth, you will have to proceed on to the next chapters to learn about z-transforms and how to apply them to the study of discrete systems.
- For digital control systems with sample rates less than 30 times bandwidth, design is often carried out directly in the discrete domain, eliminating approximation errors.

3.5 Problems

3.1 Do the following:

(a) Design a continuous lead compensation for the satellite attitude control example ($G(s) = 1/s^2$) described in Appendix A.1 so that the complex roots are at approximately $s = -4.4 \pm j4.4$ rad/sec.

(b) Assuming the compensation is to be implemented digitally, approximate the effect of the digital implementation to be a delay of $T/2$ as given by

$$G_h(s) = \frac{2/T}{s + 2/T}$$

and determine the revised root locations for sample rates of $\omega_s = 5$ Hz, 10 Hz, and 20 Hz where $T = 1/\omega_s$ sec.

3.2 Repeat Example 3.1, but use the approximation that

$$\dot{x}(k) \cong \frac{x(k) - x(k-1)}{T},$$

the **backward rectangular** version of Euler's method. Compare the resulting difference equations with the forward rectangular Euler method. Also compute the numerical value of the coefficients for both cases vs. sample rate for ω_s = 1 - 100 Hz. Assume the continuous values from Eq. (3.8). Note that the coefficients of interest are given in Eq. (3.7) for the forward rectangular case as $(1 - bT)$ and $(aT - 1)$.

3.3 For the compensation

$$D(s) = 25\frac{s+1}{s+15},$$

use Euler's forward retangular method to determine the difference equations for a digital implementation with a sample rate of 80 Hz. Repeat the calculations using the backward retangular method (see Problem 3.2) and compare the difference equation coefficients.

3.4 For the compensation

$$D(s) = 5\frac{s+2}{s+20},$$

use Euler's forward retangular method to determine the difference equations for a digital implementation with a sample rate of 80 Hz. Repeat the calculations using the backward retangular method (see Problem 3.2) and compare the difference equation coefficients.

3.5 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{1000}{s^2}.$$

Design a digital PID controller that has a bandwidth of 100 Hz, a phase margin of 50°, and has no output error for a constant bias torque from the drive motor. Use a sample rate of 6 kHz.

3.6 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{1000}{s^2}.$$

Design a digital controller that has a bandwidth of 100 Hz and a phase margin of 50°. Use a sample rate of 6 kHz.

3.7 For

$$G(s) = \frac{1}{s^2},$$

(a) design a continuous compensation so that the closed-loop system has a rise time $t_r < 1$ sec and overshoot $M_p < 15\%$ to a step input command,

(b) revise the compensation so the specifications would still be met if the feedback was implemented digitally with a sample rate of 5 Hz, and

(c) find difference equations that will implement the compensation in the digital computer.

3.8 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{500}{s^2}.$$

Design a continuous lead compensation so that the closed-loop system has a bandwidth of 100 Hz and a phase margin of 50°. Modify the MATLAB file fig32.m so that you can evaluate the digital version of your lead compensation using Euler's forward retangular method. Try different sample rates, and find the slowest one where the overshoot does not exceed 30%.

3.9 The antenna tracker has the transfer function

$$G(s) = \frac{10}{s(s+2)}.$$

Design a continuous lead compensation so that the closed-loop system has a rise time $t_r < 0.3$ sec and overshoot $M_p < 10\%$. Modify the MATLAB file fig32.m so that you can evaluate the digital version of your lead compensation using Euler's forward retangular method. Try different sample rates, and find the slowest one where the overshoot does not exceed 20%.

3.10 The antenna tracker has the transfer function

$$G(s) = \frac{10}{s(s+2)}.$$

Design a continuous lead compensation so that the closed-loop system has a rise time $t_r < 0.3$ sec and overshoot $M_p < 10\%$. Approximate the effect of a digital implementation to be

$$G_h(s) = \frac{2/T}{s+2/T},$$

and estimate M_p for a digital implementation with a sample rate of 10 Hz.