Feedback Control of Dynamic Systems

Sixth Edition

Gene F. Franklin
Stanford University

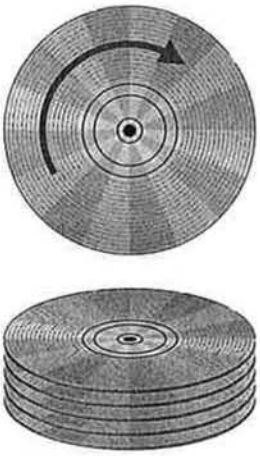
J. David Powell
Stanford University

Abbas Emami-Naeini
SC Solutions, Inc.

PEARSON

Upper Saddle River Boston Columbus San Francisco New York
Indianapolis London Toronto Sydney Singapore Tokyo Montreal
Dubai Madrid Hong Kong Mexico City Munich Paris Amsterdam Cape Town

8 Digital Control



A Perspective on Digital Control

Most of the controllers we have studied so far were described by the Laplace transform or differential equations, which, strictly speaking, are assumed to be built using analog electronics, such as those in Figs. 5.31 and 5.35. However, as discussed in Section 4.4, most control systems today use digital computers (usually microprocessors) to implement the controllers. The intent of this chapter is to expand on the design of control systems that will be implemented in a digital computer. The implementation leads to an average delay of half the sample period and to a phenomenon called aliasing, which need to be addressed in the controller design.

Analog electronics can integrate and differentiate signals. In order for a digital computer to accomplish these tasks, the differential equations describing compensation must be approximated by reducing them to algebraic equations involving addition, division, and multiplication, as developed in Section 4.4. This chapter expands on various ways to make these approximations. The resulting design can then be tuned up, if needed, using direct digital analysis and design.

You should be able to design, analyze, and implement a digital control system from the material in this chapter. However, our treatment here is a limited version of a complex subject covered in more detail in *Digital Control of Dynamic Systems* by Franklin et al. (1998 3rd ed.).

Chapter Overview

In Section 8.1 we describe the basic structure of digital control systems and introduce the issues that arise due to the sampling. The digital implementation described in Section 4.4 is sufficient for implementing a feedback control law in a digital control system, which you can then evaluate via SIMULINK® to determine the degradation with respect to the continuous case. However, to fully understand the effect of sampling, it is useful to learn about discrete linear analysis tools. This requires an understanding of the z-transform, which we discuss in Section 8.2. Section 8.3 builds on this understanding to provide a better foundation for design using discrete equivalents that was briefly discussed in Section 4.4. Hardware characteristics and sample rate issues are discussed in Sections 8.4 and 8.5, both of which need to be addressed in order to implement a digital controller.

In contrast to discrete equivalent design, which is an approximate method, optional Section 8.6 explores direct digital design (also called discrete design), which provides an exact method that is independent of whether the sample rate is fast or not.

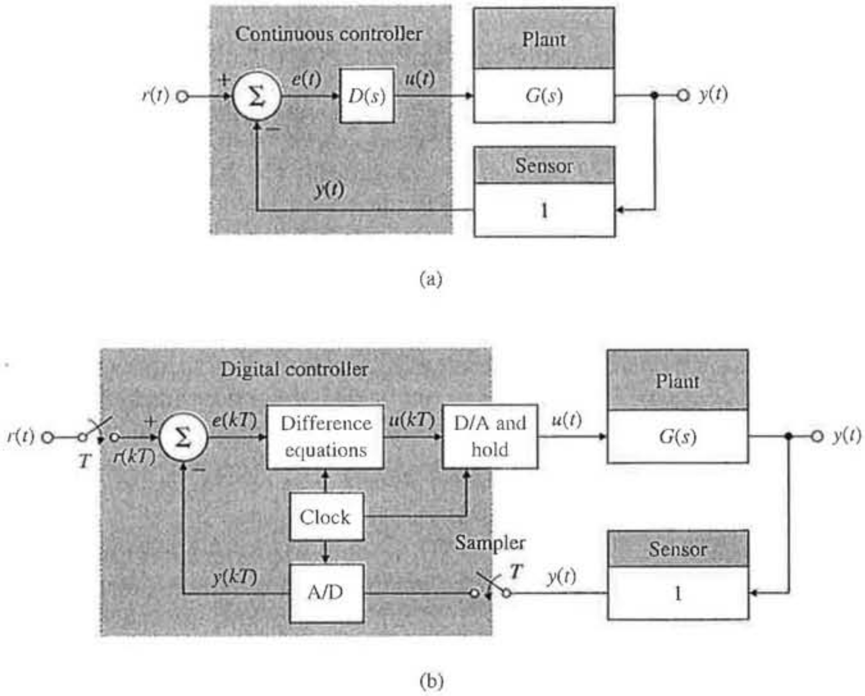
8.1 Digitization

Figure 8.1(a) shows the topology of the typical continuous system that we have been considering in previous chapters. 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. 8.1(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 control provided by $D(s)$ must be generated by algebraic recursive equations.

We consider first the action of the **analog-to-digital (A/D) converter** on a signal. This device samples a physical variable, most commonly an electrical voltage, and converts it into a binary number that usually consists of 10 to 16 bits. Conversion from the analog signal $y(t)$ to the samples, $y(kT)$, occurs repeatedly at instants of time T seconds apart. T is the **sample period**, and $1/T$ is the **sample rate** in Hertz. 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 signal such as $y(t)$, which changes continuously in time. A system having both discrete and continuous signals is called a **sampled data system**.

Sample period

We make the assumption that the sample period is fixed. In practice, digital control systems sometimes have varying sample periods and/or different periods in different feedback paths. Usually, the computer logic includes a **clock** that supplies a pulse, or **interrupt**, every T seconds, and the A/D converter sends a number to the computer each time the interrupt arrives. An alternative implementation, often referred to as **free running**, is to access the A/D converter after each cycle of code execution has been completed. In the former case the sample period is precisely fixed; in the latter



(a)

(b)

Figure 8.1
Block diagrams for a basic control system: (a) continuous system; (b) with a digital computer

case the sample period is fixed essentially by the length of the code, provided that no logic branches are present, which could vary the amount of code executed.

Zero-order hold (ZOH)

Discrete equivalents

There also may be a sampler and an A/D converter for the input command $r(t)$, which produces the discrete $r(kT)$, from which the sensed output $y(kT)$ will be subtracted to arrive at the discrete error signal $e(kT)$. As we saw in Sections 4.4 and 5.4.4, and Example 6.15, the continuous compensation is approximated by difference equations, which are the discrete version of differential equations and can be made to duplicate the dynamic behavior of $D(s)$ if the sample period is short enough. The result of the difference equations is a discrete $u(kT)$ at each sample instant. This signal is converted to a continuous $u(t)$ by the **digital-to-analog (D/A) converter** and the hold: the D/A converter changes the binary number to an analog voltage, and a **zero-order hold** 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. There are two basic techniques for finding the difference equations for the digital controller. One technique, called **discrete equivalent**, consists of designing a continuous compensation $D(s)$ using methods described in the previous chapters, then approximating that $D(s)$ using the method of Section 4.4 (Tustin's Method), or one of the other methods described in Section 8.3. The other technique is **discrete design**, described in Section 8.6. Here the difference equations are found directly without designing $D(s)$ first.

Sample rate selection

The sample rate required depends on the closed-loop bandwidth of the system. Generally, sample rates should be about 20 times the bandwidth or faster in order to assure that the digital controller will match the performance of the continuous

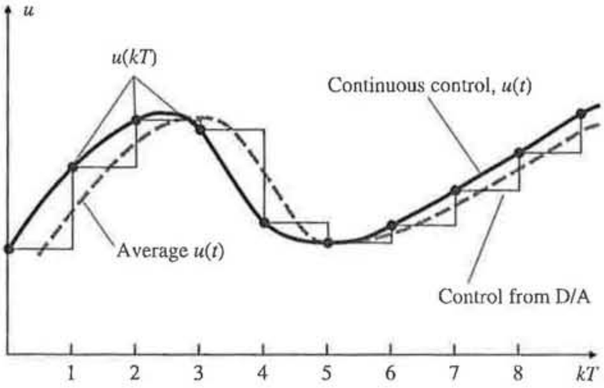


Figure 8.2 The delay due to the hold operation

controller. Slower sample rates can be used if some adjustments are made in the digital controller or some performance degradation is acceptable. Use of the discrete design method described in Section 8.6 allows for a much slower sample rate if that is desirable to minimize hardware costs; however, best performance of a digital controller is obtained when the sample rate is greater than 25 times the bandwidth.

It is worth noting that the single most important impact of implementing a control system digitally is the delay associated with the hold. Because each value of $u(kT)$ in Fig. 8.1(b) is held constant until the next value is available from the computer, the continuous value of $u(t)$ consists of steps (see Fig. 8.2) that, on average, are delayed from $u(kT)$ by $T/2$ as shown in the figure. If we simply incorporate this $T/2$ delay into a continuous analysis of the system, an excellent prediction of the effects of sampling results for sample rates much slower than 20 times bandwidth. We will discuss this further in Section 8.3.3.

8.2 Dynamic Analysis of Discrete Systems

The z-transform is the mathematical tool for the analysis of linear discrete systems. It plays the same role for discrete systems that the Laplace transform does for continuous systems. This section will give a short description of the z-transform, describe its use in analyzing discrete systems, and show how it relates to the Laplace transform.

8.2.1 z-Transform

In the analysis of continuous systems, we use the Laplace transform, which is defined by

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}\,dt,$$

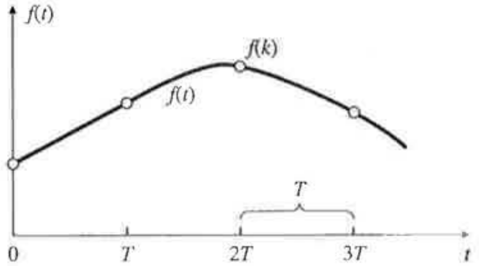
which leads directly to the important property that (with zero initial conditions)

$$\mathcal{L}\{\dot{f}(t)\} = sF(s). \tag{8.1}$$

Relation (8.1) enables us easily to find the transfer function of a linear continuous system, given the differential equation of that system.

Figure 8.3

A continuous, sampled version of signal f



z-transform

For discrete systems a similar procedure is available. The **z-transform** is defined by

$$\mathcal{Z}\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}, \tag{8.2}$$

where $f(k)$ is the sampled version of $f(t)$, as shown in Fig. 8.3, and $k = 0, 1, 2, 3, \ldots$ refers to discrete sample times $t_0, t_1, t_2, t_3, \ldots$. This leads directly to a property analogous to Eq. (8.1), specifically, that

$$\mathcal{Z}\{f(k-1)\} = z^{-1}F(z). \tag{8.3}$$

This relation allows us to easily find the transfer function of a discrete system, given the difference equations of that system. For example, the general second-order difference equation

$$y(k) = -a_1y(k-1) - a_2y(k-2) + b_0u(k) + b_1u(k-1) + b_2u(k-2)$$

can be converted from this form to the z-transform of the variables $y(k)$, $u(k)$, ... by invoking Eq. (8.3) once or twice to arrive at

$$Y(z) = (-a_1z^{-1} - a_2z^{-2})Y(z) + (b_0 + b_1z^{-1} + b_2z^{-2})U(z). \tag{8.4}$$

Discrete transfer function

Equation (8.4) then results in the discrete transfer function

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}.$$

8.2.2 z-Transform Inversion

Table 8.1 relates simple discrete-time functions to their z-transforms and gives the Laplace transforms for the same time functions.

Given a general z-transform, we could expand it into a sum of elementary terms using partial-fraction expansion (see Appendix A) and find the resulting time series from the table. These procedures are exactly the same as those used for continuous systems; as with the continuous case, most designers would use a numerical evaluation of the discrete equations to obtain a time history rather than inverting the z-transform.

A z-transform inversion technique that has no continuous counterpart is called **long division**. Given the z-transform

$$Y(z) = \frac{N(z)}{D(z)}, \tag{8.5}$$

TABLE 8.1 **Laplace Transforms and *z*-Transforms of Simple Discrete-Time Functions**

No.	$F(s)$	$f(kT)$	$F(z)$
1		$1, k=0; 0, k \neq 0$	1
2		$1, k=k_o; 0, k \neq k_o$	z^{-k_o}
3	$\frac{1}{s}$	$1(kT)$	$\frac{z}{z-1}$
4	$\frac{1}{s^2}$	kT	$\frac{Tz}{(z-1)^2}$
5	$\frac{1}{s^3}$	$\frac{1}{2!}(kT)^2$	$\frac{T^2}{2}\left[\frac{z(z+1)}{(z-1)^3}\right]$
6	$\frac{1}{s^4}$	$\frac{1}{3!}(kT)^3$	$\frac{T^3}{6}\left[\frac{z(z^2+4z+1)}{(z-1)^4}\right]$
7	$\frac{1}{s^m}$	$\lim_{a\to 0}\frac{(-1)^{m-1}}{(m-1)!}\left(\frac{\partial^{m-1}}{\partial a^{m-1}}e^{-akT}\right)$	$\lim_{a\to 0}\frac{(-1)^{m-1}}{(m-1)!}\left(\frac{\partial^{m-1}}{\partial a^{m-1}}\frac{z}{z-e^{-aT}}\right)$
8	$\frac{1}{s+a}$	e^{-akT}	$\frac{z}{z-e^{-aT}}$
9	$\frac{1}{(s+a)^2}$	kTe^{-akT}	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
10	$\frac{1}{(s+a)^3}$	$\frac{1}{2}(kT)^2e^{-akT}$	$\frac{T^2}{2}e^{-aT}z\frac{(z+e^{-aT})}{(z-e^{-aT})^3}$
11	$\frac{1}{(s+a)^m}$	$\frac{(-1)^{m-1}}{(m-1)!}\left(\frac{\partial^{m-1}}{\partial a^{m-1}}e^{-akT}\right)$	$\frac{(-1)^{m-1}}{(m-1)!}\left(\frac{\partial^{m-1}}{\partial a^{m-1}}\frac{z}{z-e^{-aT}}\right)$
12	$\frac{a}{s(s+a)}$	$1-e^{-akT}$	$\frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$
13	$\frac{a}{s^2(s+a)}$	$\frac{1}{a}(akT-1+e^{-akT})$	$\frac{z[(aT-1+e^{-aT})z+(1-e^{-aT}-aTe^{-aT})]}{a(z-1)^2(z-e^{-aT})}$
14	$\frac{b-a}{(s+a)(s+b)}$	$e^{-akT}-e^{-bkT}$	$\frac{(e^{-aT}-e^{-bT})z}{(z-e^{-aT})(z-e^{-bT})}$
15	$\frac{s}{(s+a)^2}$	$(1-akT)e^{-akT}$	$\frac{z[z-e^{-aT}(1+aT)]}{(z-e^{-aT})^2}$
16	$\frac{a^2}{s(s+a)^2}$	$1-e^{-akT}(1+akT)$	$\frac{z[z(1-e^{-aT}-aTe^{-aT})+e^{-2aT}-e^{-aT}+aTe^{-aT}]}{(z-1)(z-e^{-aT})^2}$
17	$\frac{(b-a)s}{(s+a)(s+b)}$	$be^{-bkT}-ae^{-akT}$	$\frac{z[z(b-a)-(be^{-aT}-ae^{-bT})]}{(z-e^{-aT})(z-e^{-bT})}$
18	$\frac{a}{s^2+a^2}$	$\sin akT$	$\frac{z\sin aT}{z^2-(2\cos aT)z+1}$
19	$\frac{s}{s^2+a^2}$	$\cos akT$	$\frac{z(z-\cos aT)}{z^2-(2\cos aT)z+1}$
20	$\frac{s+a}{(s+a)^2+b^2}$	$e^{-akT}\cos bkT$	$\frac{z(z-e^{-aT}\cos bT)}{z^2-2e^{-aT}(\cos bT)z+e^{-2aT}}$
21	$\frac{b}{(s+a)^2+b^2}$	$e^{-akT}\sin bkT$	$\frac{ze^{-aT}\sin bT}{z^2-2e^{-aT}(\cos bT)z+e^{-2aT}}$
22	$\frac{a^2+b^2}{s[(s+a)^2+b^2]}$	$1-e^{-akT}(\cos bkT+\frac{a}{b}\sin bkT)$	$\frac{z(Az+B)}{(z-1)[z^2-2e^{-aT}(\cos bT)z+e^{-2aT}]}$ $A=1-e^{-aT}\cos bT-\frac{a}{b}e^{-aT}\sin bT$ $B=e^{-2aT}+\frac{a}{b}e^{-aT}\sin bT-e^{-aT}\cos bT$

$F(s)$ is the Laplace transform of $f(t)$, and $F(z)$ is the z-transform of $f(kT)$.
Note: $f(t)=0$ for $t=0$.

z-transform inversion: long division

we simply divide the denominator into the numerator using long division. The result is a series (perhaps with an infinite number of terms) in z^{-1}, from which the time series can be found by using Eq. (8.2).

For example, a first-order system described by the difference equation

$$y(k) = \alpha y(k-1) + u(k)$$

yields the discrete transfer function

$$\frac{Y(z)}{U(z)} = \frac{1}{1-\alpha z^{-1}}.$$

For a unit-pulse input defined by

$$\begin{aligned} u(0) &= 1, \\ u(k) &= 0 \quad k \neq 0, \end{aligned}$$

the z-transform is then

$$U(z) = 1, \tag{8.6}$$

so

$$Y(z) = \frac{1}{1-\alpha z^{-1}}. \tag{8.7}$$

Therefore, to find the time series, we divide the numerator of Eq. (8.7) by its denominator using long division:

$$\begin{array}{r|l} & 1 + \alpha z^{-1} + \alpha^2 z^{-2} + \alpha^3 z^{-3} + \cdots \\ \hline 1-\alpha z^{-1} & 1 \\ & 1 - \alpha z^{-1} \\ \hline & \quad \alpha z^{-1} + 0 \\ & \quad \alpha z^{-1} - \alpha^2 z^{-2} \\ \hline & \qquad \alpha^2 z^{-2} + 0 \\ & \qquad \alpha^2 z^{-2} - \alpha 3 z^{-3} \\ \hline & \qquad\qquad \alpha^3 z^{-3} \\ & \qquad\qquad\quad \ddots \end{array}$$

This yields the infinite series

$$Y(z) = 1 + \alpha z^{-1} + \alpha^2 z^{-2} + \alpha^3 z^{-3} + \cdots. \tag{8.8}$$

From Eqs. (8.8) and (8.2) we see that the sampled time history of y is

$$\begin{aligned} y(0) &= 1, \\ y(1) &= \alpha, \\ y(2) &= \alpha^2, \\ \vdots\quad & \quad\vdots \\ y(k) &= \alpha^k. \end{aligned}$$

8.2.3 Relationship between *s* and *z*

For continuous systems, we saw in Chapter 3 that certain behaviors result from different pole locations in the s-plane: oscillatory behavior for poles near the imaginary axis, exponential decay for poles on the negative real axis, and unstable behavior for poles with a positive real part. A similar kind of association would also be useful to know when designing discrete systems. Consider the continuous signal

$$f(t) = e^{-at}, \quad t > 0,$$

which has the Laplace transform

$$F(s) = \frac{1}{s+a}$$

and corresponds to a pole at $s = -a$. The z-transform of $f(kT)$ is

$$F(z) = \mathcal{Z}\{e^{-akT}\}. \tag{8.9}$$

From Table 8.1 we can see that Eq. (8.9) is equivalent to

$$F(z) = \frac{z}{z - e^{-aT}},$$

which corresponds to a pole at $z = e^{-aT}$. This means that a pole at $s = -a$ in the s-plane corresponds to a pole at $z = e^{-aT}$ in the discrete domain. This is true in general:

Relationship between *z*-plane and *s*-plane characteristics

> The equivalent characteristics in the z-plane are related to those in the s-plane by the expression
>
> $$z = e^{sT}, \tag{8.10}$$
>
> where T is the sample period.

Table 8.1 also includes the Laplace transforms, which demonstrates the $z = e^{sT}$ relationship for the roots of the denominators of the table entries for $F(s)$ and $F(z)$.

Figure 8.4 shows the mapping of lines of constant damping ζ and natural frequency ω_n from the s-plane to the upper half of the z-plane, using Eq. (8.10). The mapping has several important features (see Problem 8.4):

1. The stability boundary is the unit circle $|z| = 1$.
2. The small vicinity around $z = +1$ in the z-plane is essentially identical to the vicinity around $s = 0$ in the s-plane.
3. The z-plane locations give response information normalized to the sample rate, rather than to time as in the s-plane.
4. The negative real z-axis always represents a frequency of $\omega_s/2$, where $\omega_s = 2\pi/\mathrm{T} =$ sample rate in radians per second.
5. Vertical lines in the left half of the s-plane (the constant real part or time constant) map into circles within the unit circle of the z-plane.
6. Horizontal lines in the s-plane (the constant imaginary part of the frequency) map into radial lines in the z-plane.

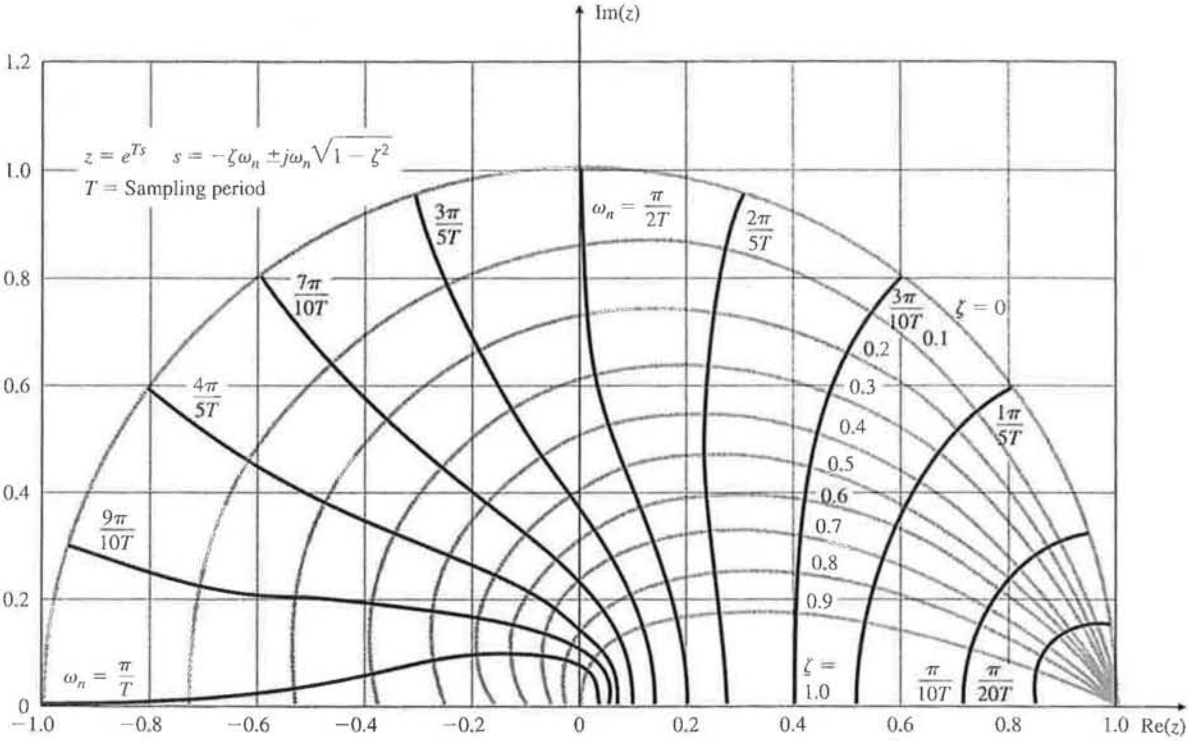


Figure 8.4

Natural frequency (solid color) and damping loci (light color) in the z-plane; the portion below the Re(z)-axis (not shown) is the mirror image of the upper half shown

Nyquist frequency $= \omega_S/2$

7. Frequencies greater than $\omega_s/2$, called the **Nyquist frequency**, appear in the z-plane on top of corresponding lower frequencies because of the circular character of the trigonometric functions imbedded in Eq. (8.10). This overlap is called **aliasing** or **folding**. As a result it is necessary to sample at least twice as fast as a signal's highest frequency component in order to represent that signal with the samples. (We will discuss aliasing in greater detail in Section 8.4.3.)

   To provide insight into the correspondence between z-plane locations and the resulting time sequence, Fig. 8.5 sketches time responses that would result from poles at the indicated locations. This figure is the discrete companion of Fig. 3.15.

8.2.4 Final Value Theorem

The Final Value Theorem for continuous systems, which we discussed in Section 3.1.6, states that

$$\lim_{t\to\infty} x(t) = x_{ss} = \lim_{s\to 0} sX(s), \tag{8.11}$$

as long as all the poles of $sX(s)$ are in the left half-plane (LHP). It is often used to find steady-state system errors and/or steady-state gains of portions of a control

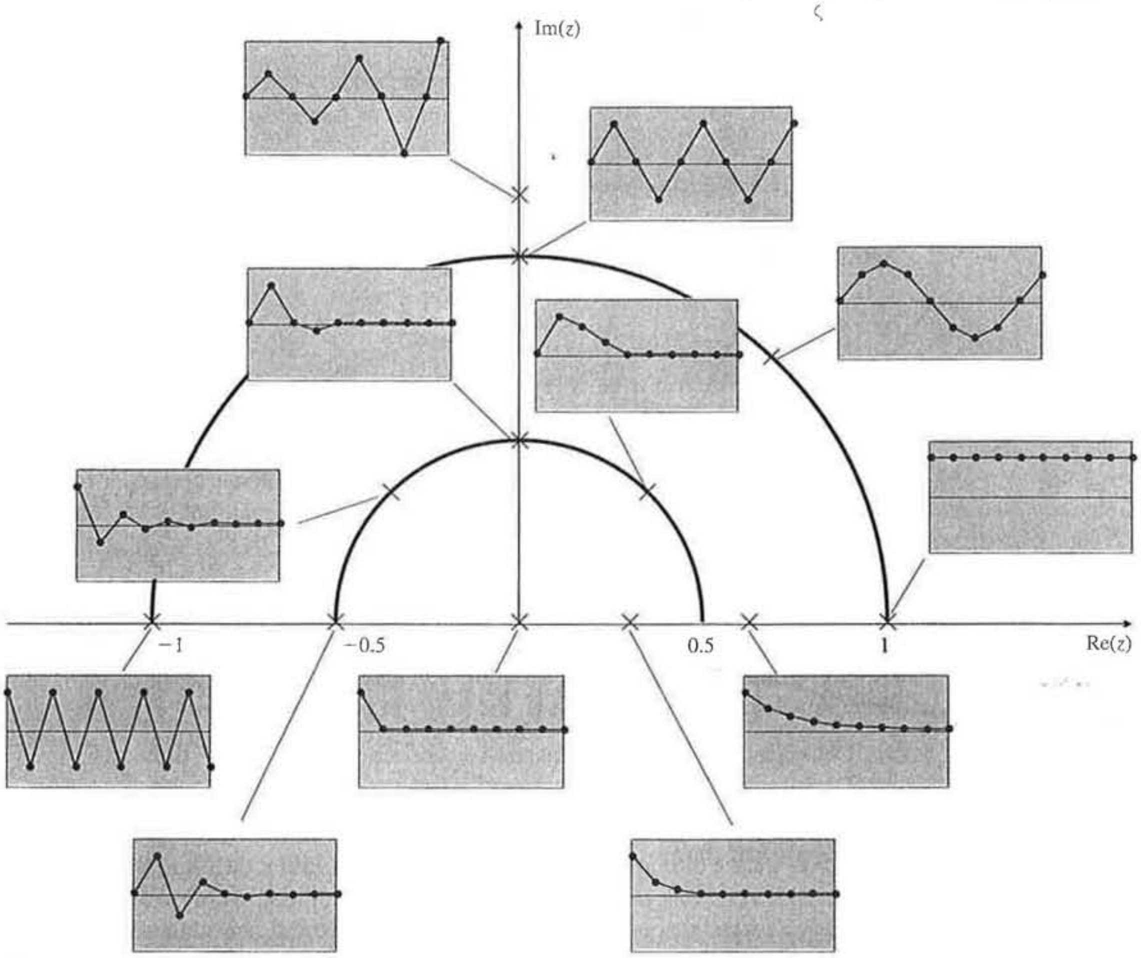


Figure 8.5

Time sequences associated with points in the z-plane

system. We can obtain a similar relationship for discrete systems by noting that a constant continuous steady-state response is denoted by $X(s) = {}^{A}/_{s}$ and leads to the multiplication by s in Eq. (8.11). Therefore, because the constant steady-state response for discrete systems is

$$X(z) = \frac{A}{1 - z^{-1}},$$

Final Value Theorem for discrete systems

the discrete Final Value Theorem is

$$\lim_{k \to \infty} x(k) = x_{ss} = \lim_{z \to 1} (1 - z^{-1}) X(z) \tag{8.12}$$

if all the poles of $(1 - z^{-1})X(z)$ are inside the unit circle.

For example, to find the DC gain of the transfer function

$$G(z) = \frac{X(z)}{U(z)} = \frac{0.58(1+z)}{z + 0.16},$$

we let $u(k) = 1$ for $k \geq 0$, so that

$$U(z) = \frac{1}{1 - z^{-1}}$$

and

$$X(z) = \frac{0.58(1 + z)}{(1 - z^{-1})(z + 0.16)}.$$

Applying the Final Value Theorem yields

$$x_{ss} = \lim_{z \to 1} \left[\frac{0.58(1 + z)}{z + 0.16} \right] = 1,$$

DC gain

so the DC gain of $G(z)$ is unity. To find the DC gain of any stable transfer function, we simply substitute $z = 1$ and compute the resulting gain. Because the DC gain of a system should not change whether represented continuously or discretely, this calculation is an excellent aid to check that an equivalent discrete controller matches a continuous controller. It is also a good check on the calculations associated with determining the discrete model of a system.

8.3 Design Using Discrete Equivalents

Stages in design using discrete equivalents

Design by discrete equivalent, sometimes called **emulation**, is partially described in Section 4.4 and proceeds through the following stages:

1. Design a continuous compensation as described in Chapters 1 through 7.
2. Digitize the continuous compensation.
3. Use discrete analysis, simulation, or experimentation to verify the design.

In Section 4.4 we discussed Tustin's method for performing the digitization. Armed with an understanding of the z-transform from Section 8.2, we now develop more digitization procedures and analyze the performance of the digitally controlled system.

Assume that we are given a continuous compensation $D(s)$ as shown in Fig. 8.1(a). We wish to find a set of difference equations or $D(z)$ for the digital implementation of that compensation in Fig. 8.1(b). First we rephrase the problem as one of finding the best $D(z)$ in the digital implementation shown in Fig. 8.6(a) to match the continuous system represented by $D(s)$ in Fig. 8.6(b). In this section we examine and compare three methods for solving this problem.

It is important to remember, as stated earlier, that these methods are approximations; there is no exact solution for all possible inputs because $D(s)$ responds to the complete time history of $e(t)$, whereas $D(z)$ has access to only the samples $e(kT)$. In a sense, the various digitization techniques simply make different assumptions about what happens to $e(t)$ between the sample points.

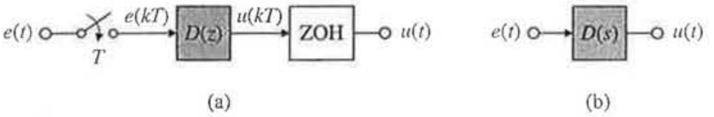


Figure 8.6
Comparison of (a) digital and; (b) continuous implementation

Tustin's Method

As discussed in Section 4.4, one digitization technique is to approach the problem as one of numerical integration. Suppose

$$\frac{U(s)}{E(s)} = D(s) = \frac{1}{s},$$

which is integration. Therefore,

$$u(kT) = \int_0^{kT-T} e(t)\,dt + \int_{kT-T}^{kT} e(t)\,dt, \tag{8.13}$$

which can be rewritten as

$$u(kT) = u(kT - T) + \text{area under } e(t) \text{ over last } T, \tag{8.14}$$

where T is the sample period.

For Tustin's method, the task at each step is to use trapezoidal integration, that is, to approximate $e(t)$ by a straight line between the two samples (Fig. 8.7). Writing $u(kT)$ as $u(k)$ and $u(kT - T)$ as $u(k-1)$ for short, we convert Eq. (8.14) to

$$u(k) = u(k-1) + \frac{T}{2}[e(k-1) + e(k)], \tag{8.15}$$

or, taking the z-transform,

$$\frac{U(z)}{E(z)} = \frac{T}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right) = \frac{1}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}. \tag{8.16}$$

For $D(s) = a/(s+a)$, applying the same integration approximation yields

$$D(z) = \frac{a}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + a}.$$

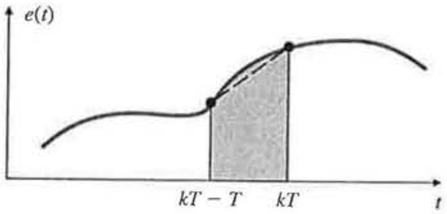
In fact, substituting

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

Tustin's method or bilinear approximation

for every occurrence of s in any $D(s)$ yields a $D(z)$ based on the trapezoidal integration formula. This is called **Tustin's method** or the **bilinear approximation**. Finding Tustin's approximation by hand for even a simple transfer function requires fairly extensive algebraic manipulations. The c2d function of MATLAB® expedites the process, as shown in the next example.

Figure 8.7
Trapezoidal integration



EXAMPLE 8.1 *Digital Controller for Example 6.15 Using Tustin's Approximation*

Determine the difference equations to implement the compensation from Example 6.15,

$$D(s) = 10\frac{s/2+1}{s/10+1},$$

at a sample rate of 25 times bandwidth using Tustin's approximation. Compare the performance against the continuous system and the discrete implementation done in Example 6.15 at a slower sample rate.

Solution. The bandwidth (ω_{BW}) for Example 6.15 is approximately 10 rad/sec, as can be deduced by observing that the crossover frequency (ω_c) is approximately 5 rad/sec and noting the relationship between ω_c and ω_{BW} in Fig. 6.51. Therefore, the sample frequency should be

$$\omega_s = 25 \times \omega_{BW} = (25)(10) = 250 \text{ rad/sec.}$$

Normally, when a frequency is indicated with the units of cycles per second, or Hz, it is given the symbol f, so with this convention, we have

$$f_s = \omega_s/(2\pi) \simeq 40 \text{ Hz}, \tag{8.17}$$

and the sample period is then

$$T = 1/f_s = 1/40 = 0.025 \text{ sec.}$$

The discrete compensation is computed by the MATLAB statement

```
sysDs = tf(10*[0.5 1],[0.1 1]);
sysDd = c2d(sysDs,0.025,'tustin');
```

which produces

$$D(z) = 10\frac{4.556 - 4.333\,z^{-1}}{1 - 0.7778\,z^{-1}}. \tag{8.18}$$

We can then write the difference equation by inspecting Eq. (8.18) to get

$$u(k) = 0.7778u(k-1) + 45.56e(k) - 43.33e(k-1),$$

or, indexing all time variables by 1, the equivalent is

$$u(k+1) = 0.7778u(k) + 45.56[e(k+1) - 0.9510e(k)]. \tag{8.19}$$

Equation (8.19) computes 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 $e(k)$.

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. The instructions to the computer to implement the feedback loop in Fig. 8.1(b) with the difference equation from Eq. (8.19) would call for a continual looping through the following code:

```
READ $y$, $r$
    $e = r - y$
    $u = 0.7778u_p + 45.56\left[e - 0.9510e_p\right]$
```

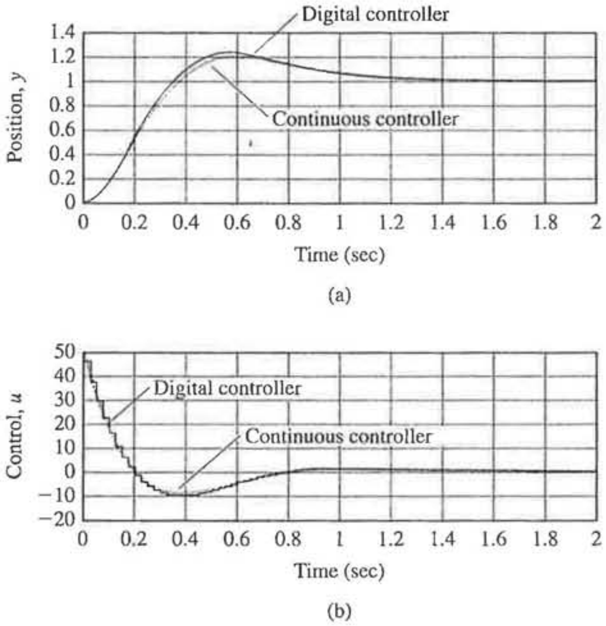


Figure 8.8
Comparison between the digital and the continuous controller step response with a sample rate 25 times bandwidth:
(a) position; (b) control

```
OUTPUT $u$
    $u_p = u$ (where $u_p$ will be the past value for the next loop through)
    $e_p = e$
go back to READ when $T$ sec have elapsed since last READ
```

Use of SIMULINK to compare the two implementations, in a manner similar to that used for Example 6.15, yields the step responses shown in Fig. 8.8. Note that sampling at 25 times bandwidth causes the digital implementation to match the continuous one quite well. Also note that the same case with half the sampling rate whose step response is shown in Fig. 6.59 contains a noticeable degradation in the overshoot (and damping) compared to the continuous case. Generally speaking, if you want to match a continuous system with a digital approximation of the continuous compensation, it is wise to sample at approximately 25 times bandwidth or faster.

8.3.1 Matched Pole–Zero (MPZ) Method

Another digitization method, called the **matched pole–zero** method, is found by extrapolating from the relationship between the s- and z-planes stated in Eq. (8.10). If we take the z-transform of a sampled function $x(k)$, then the poles of $X(z)$ are related to the poles of $X(s)$ according to the relation $z = e^{sT}$. The MPZ technique applies the relation $z = e^{sT}$ to the poles and zeros of a transfer function, even though, strictly speaking, this relation applies neither to transfer functions nor even to the zeros of a time sequence. Like all transfer-function digitization methods, the MPZ method is an approximation; here the approximation is motivated partly by the fact that $z = e^{sT}$ is the correct s to z transformation for the poles of the transform of a time sequence and partly by the minimal amount of algebra required to determine the digitized transfer function by hand, so as to facilitate checking the computer calculations.

Because physical systems often have more poles than zeros, it is useful to arbitrarily add zeros at $z = -1$, resulting in a $1 + z^{-1}$ term in $D(z)$. This causes an averaging of the current and past input values, as in Tustin's method. We select the low-frequency gain of $D(z)$ so that it equals that of $D(s)$.

MPZ Method Summary

1. Map poles and zeros according to the relation $z = e^{sT}$.
2. If the numerator is of lower order than the denominator, add powers of $(z + 1)$ to the numerator until numerator and denominator are of equal order.
3. Set the DC or low-frequency gain of $D(z)$ equal to that of $D(s)$.

The MPZ approximation of

$$D(s) = K_c \frac{s+a}{s+b} \tag{8.20}$$

is

$$D(z) = K_d \frac{z - e^{-aT}}{z - e^{-bT}}, \tag{8.21}$$

where K_d is found by causing the DC gain of $D(z)$ to equal the DC gain of $D(s)$ using the continuous and discrete versions of the Final Value Theorem. The result is

$$K_c \frac{a}{b} = K_d \frac{1 - e^{-aT}}{1 - e^{-bT}},$$

or

$$K_d = K_c \frac{a}{b} \left(\frac{1 - e^{-bT}}{1 - e^{-aT}} \right). \tag{8.22}$$

For a $D(s)$ with a higher-order denominator, Step 2 in the method calls for adding the $(z + 1)$ term. For example,

$$D(s) = K_c \frac{s+a}{s(s+b)} \Rightarrow D(z) = K_d \frac{(z+1)(z - e^{-aT})}{(z-1)(z - e^{-bT})}, \tag{8.23}$$

where, after dropping the poles at $s = 0$ and $z = 1$,

$$K_d = K_c \frac{a}{2b} \left(\frac{1 - e^{-bT}}{1 - e^{-aT}} \right). \tag{8.24}$$

In the digitization methods described so far, the same power of z appears in the numerator and denominator of $D(z)$. This implies that the difference equation output at time k will require a sample of the input at time k. For example, the $D(z)$ in Eq. (8.21) can be written

$$\frac{U(z)}{E(z)} = D(z) = K_d \frac{1 - \alpha z^{-1}}{1 - \beta z^{-1}}, \tag{8.25}$$

where $\alpha = e^{-aT}$ and $\beta = e^{-bT}$. By inspection we can see that Eq. (8.25) results in the difference equation

$$u(k) = \beta u(k-1) + K_d[e(k) - \alpha e(k-1)]. \tag{8.26}$$

EXAMPLE 8.2

Design of a Space Station Attitude Digital Controller Using Discrete Equivalents

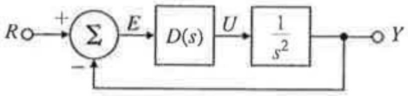
A very simplified model of the space station attitude control dynamics has the plant transfer function

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

Design a digital controller to have a closed-loop natural frequency $\omega_n \cong 0.3$ rad/sec and a damping ratio $\zeta = 0.7$.

Figure 8.9

Continuous-design definition for Example 8.2



Solution. The first step is to find the proper $D(s)$ for the system defined in Fig. 8.9. After some trial and error, we find that the specifications can be met by the lead compensation

$$D(s) = 0.81\frac{s+0.2}{s+2}. \tag{8.27}$$

The root locus in Fig. 8.10 verifies the appropriateness of using Eq. (8.27).

To digitize this $D(s)$, we first need to select a sample rate. For a system with $\omega_n = 0.3$ rad/sec, the bandwidth will also be about 0.3 rad/sec, and an acceptable sample rate would be about 20 times ω_n. Thus

$$\omega_s = 0.3 \times 20 = 6 \text{ rad/sec.}$$

A sample rate of 6 rad/sec is about 1 Hertz; therefore, the sample period should be $T = 1$ sec. The MPZ digitization of Eq. (8.27), given by Eqs. (8.21) and (8.22), yields

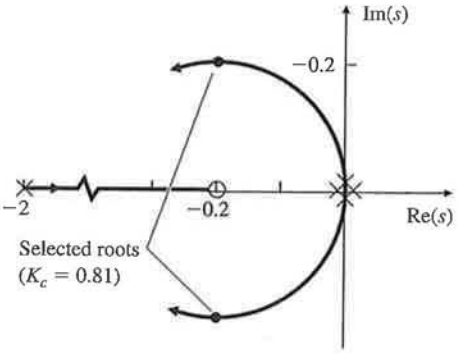
$$\begin{aligned} D(z) &= 0.389\frac{z-0.82}{z-0.135} \\ &= \frac{0.389-0.319z^{-1}}{1-0.135z^{-1}}. \end{aligned} \tag{8.28}$$

Inspection of Eq. (8.28) gives us the difference equation

$$u(k) = 0.135u(k-1) + 0.389e(k) - 0.319e(k-1), \tag{8.29}$$

Figure 8.10

s-Plane locus with respect to K



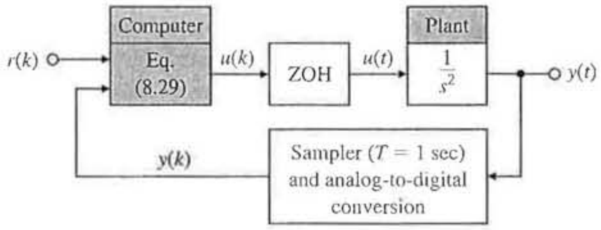


Figure 8.11
A digital control system that is equivalent to Fig. 8.9

where

$$e(k) = r(k) - y(k),$$

and this completes the digital algorithm design. The complete digital system is shown in Fig. 8.11.

The last step in the design process is to verify the design by implementing it on the computer. Figure 8.12 compares the step response of the digital system using $T = 1$ sec with the step response of the continuous compensation. Note that there is greater overshoot and a longer settling time in the digital system, which suggests a decrease in the damping. The average $^T/_2$ delay shown in Fig. 8.2 is the cause of the reduced damping. For a better match to the continuous system, it may be prudent to increase the sample rate. Figure 8.12 also shows the response with sampling that is twice as fast and it can be seen that it comes much closer to the continuous system. Note that the discrete compensation needs to be recalculated for this faster sample rate according to Eqs. (8.21) and (8.22).

It is impossible to sample $e(k)$, compute $u(k)$, and then output $u(k)$ all in zero elapsed time; therefore, Eqs. (8.26) and (8.29) are impossible to implement precisely. However, if the equation is simple enough and/or the computer is fast enough, a slight computational delay between the $e(k)$ sample and the $u(k)$ output will have a

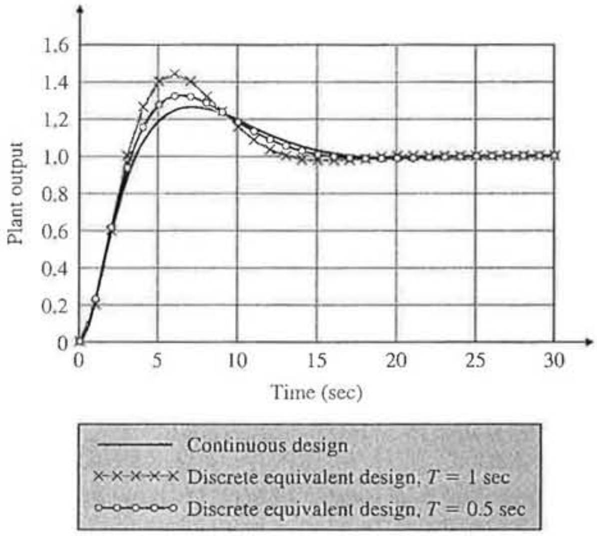


Figure 8.12
Step responses of the continuous and digital implementations

negligible effect on the actual response of the system compared with that expected from the original design. A rule of thumb would be to keep the computational delay on the order of $^1/_{10}$ of T. The real-time code and hardware can be structured so that this delay is minimized by making sure that computations between read A/D and write D/A are minimized and that $u(k)$ is sent to the ZOH immediately after its calculation.

8.3.2 Modified Matched Pole–Zero (MMPZ) Method

The $D(z)$ in Eq. (8.23) would also result in $u(k)$ being dependent on $e(k)$, the input at the same time point. If the structure of the computer hardware prohibits this relation or if the computations are particularly lengthy, it may be desirable to derive a $D(z)$ that has one less power of z in the numerator than in the denominator; hence, the computer output $u(k)$ would require only input from the previous time, that is, $e(k-1)$. To do this, we simply modify Step 2 in the matched pole–zero procedure so that the numerator is of lower order than the denominator by 1. For example, if

$$D(s) = K_c \frac{s+a}{s(s+b)},$$

we skip Step 2 to get

$$D(z) = K_d \frac{z - e^{-aT}}{(z-1)(z-e^{-bT})}, \tag{8.30}$$

$$K_d = K_c \frac{a}{b}\left(\frac{1-e^{-bT}}{1-e^{-aT}}\right).$$

To find the difference equation, we multiply the top and bottom of Eq. (8.30) by z^{-2} to obtain

$$D(z) = K_d \frac{z^{-1}(1-e^{-aT}z^{-1})}{1 - z^{-1}(1+e^{-bT}) + z^{-2}e^{-bT}}. \tag{8.31}$$

By inspecting Eq. (8.31) we can see that the difference equation is

$$u(k) = (1+e^{-bT})u(k-1) - e^{-bT}u(k-2) + K_d[e(k-1) - e^{-aT}e(k-2)].$$

In this equation an entire sample period is available to perform the calculation and to output $u(k)$, because it depends only on $e(k-1)$. A discrete analysis of this controller would therefore more accurately explain the behavior of the actual system. However, because this controller is using data that are one cycle old, it will typically not perform as well as the MPZ controller in terms of the deviations of the desired system output in the presence of random disturbances.

8.3.3 Comparison of Digital Approximation Methods

A numerical comparison of the magnitude of the frequency response for a first-order lag,

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

is made in Fig. 8.13 for the three approximation techniques at two different sample rates. The results of the $D(z)$ computations used in Fig. 8.13 are shown in Table 8.2.

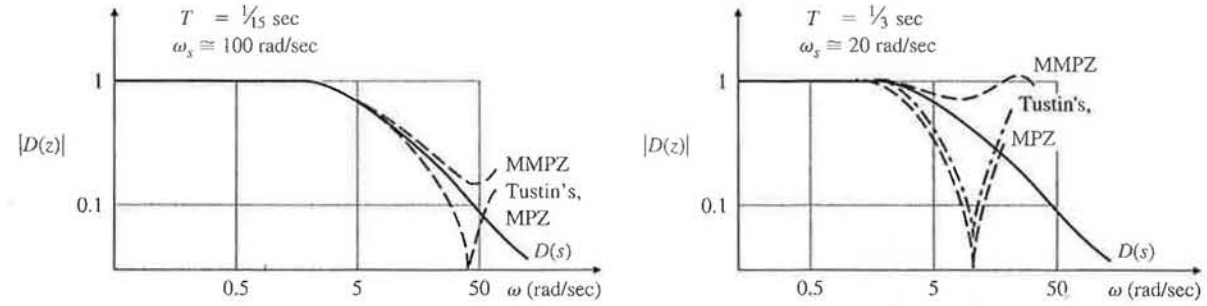


Figure 8.13
A comparison of the frequency response of three discrete approximations

Figure 8.13 shows that all the approximations are quite good at frequencies below about $1/4$ the sample rate, or $\omega_s/4$. If $\omega_s/4$ is sufficiently larger than the filter break-point frequency—that is, if the sampling is fast enough—the break-point characteristics of the lag will be accurately reproduced. Tustin's technique and the MPZ method show a notch at $\omega_s/2$ because of their zero at $z = -1$ from the $z + 1$ term. Other than the large difference at $\omega_s/2$, which is typically outside the range of interest, the three methods have similar accuracies.

8.3.4 Applicability Limits of the Discrete Equivalent Design Method

If we performed an exact discrete analysis or a simulation of a system and determined the digitization for a wide range of sample rates, the system would often be unstable for rates slower than approximately $5\omega_n$, and the damping would be degraded significantly for rates slower than about $10\omega_n$. At sample rates $\gtrsim 20\omega_n$ (or $\gtrsim 20$ times the bandwidth for more complex systems), design by discrete equivalents yields reasonable results, and at sample rates of 30 times the bandwidth or higher, discrete equivalents can be used with confidence.

ZOH transfer function

As shown by Fig. 8.2, the errors come about because the technique ignores the lagging effect of the ZOH which, on the average, is $T/2$. A method to account for

TABLE 8.2 **Comparing Digital Approximations of $D(z)$ for $D(s) = 5/(s+5)$**

	ω_s	
Method	**100 rad/sec**	**20 rad/sec**
Matched pole–zero (MPZ)	$0.143\dfrac{z+1}{z-0.715}$	$0.405\dfrac{z+1}{z-0.189}$
Modified MPZ (MMPZ)	$0.285\dfrac{1}{z-0.715}$	$0.811\dfrac{1}{z-0.189}$
Tustin's	$0.143\dfrac{z+1}{z-0.713}$	$0.454\dfrac{z+1}{z-0.0914}$

this is to approximate the $T/2$ delay with Eq. (5.94) by including a transfer function approximation for the ZOH:[1]

$$G_{\mathrm{ZOH}}(s) = \frac{2/T}{s + 2/T}. \tag{8.32}$$

Once an initial design is carried out and the sampling rate has been selected, we could improve on our discrete design by inserting Eq. (8.32) into the original plant model and adjusting the $D(s)$ so that a satisfactory response in the presence of the sampling delay is achieved. Therefore, we see that use of Eq. (8.32) partially alleviates the approximate nature of using discrete equivalents.

For sample rates slower than about $10\omega_n$ it is advisable to analyze the entire system using an exact discrete analysis. If a discrete analysis shows an unacceptable degradation of performance due to the sampling, the design can then be refined using exact discrete methods. We cover this approach in Section 8.6.

8.4 Hardware Characteristics

A digital control system includes several unique components not found in continuous control systems: an **analog-to-digital converter** is a device to sample the continuous signal voltage from the sensor and to convert that signal to a digital word; a **digital-to-analog converter** is a device to convert the digital word from the computer to an analog voltage, an **anti-alias prefilter** is an analog device designed to reduce the effects of aliasing, and the **computer** is the device where the compensation $D(z)$ is programmed and the calculations are carried out. This section provides a brief description of each of these.

8.4.1 Analog-to-Digital (A/D) Converters

As discussed in Section 8.1, A/D converters are devices that convert a voltage level from a sensor to a digital word usable by the computer. At the most basic level, all digital words are binary numbers consisting of many bits that are set to either 1 or 0. Therefore, the task of the A/D converter at each sample time is to convert a voltage level to the correct bit pattern and often to hold that pattern until the next sample time.

Of the many A/D conversion techniques that exist, the most common are based on counting schemes or a successive-approximation technique. In counting methods the input voltage may be converted to a train of pulses whose frequency is proportional to the voltage level. The pulses are then counted over a fixed period using a binary counter, thus resulting in a binary representation of the voltage level. A variation on this scheme is to start the count simultaneously with a voltage that is linear in time and to stop the count when the voltage reaches the magnitude of the input voltage to be converted.

The successive-approximation technique tends to be much faster than the counting methods. It is based on successively comparing the input voltage to reference levels representing the various bits in the digital word. The input voltage is first compared with a reference value that is half the maximum. If the input voltage is greater, the most significant bit is set, and the signal is then compared with a reference level

[1] Or other Padé approximate as discussed in Section 5.6.3

that is $^3/_4$ the maximum to determine the next bit, and so on. One clock cycle is required to set each bit, so an n-bit converter would require n cycles. At the same clock rate a counter-based converter might require as many as 2^n cycles, which would usually be much slower.

With either technique, the greater the number of bits, the longer it will take to perform the conversion. The price of A/D converters generally goes up with both speed and bit size. In 2009, a 14-bit (resolution of 0.006%) converter with a high performance capability of a 10-n sec conversion time (100 million samples per sec) sold for approximately $25 while a 12-bit (0.025%) converter with a good performance capability of a 1 μ sec conversion time (1 million samples per sec) sold for approximately $4. An 8-bit (0.4% resolution) with a 1 μ sec conversion time sold for approximately $1. The performance has been improving considerably every year.

If more than one channel of data needs to be sampled and converted to digital words, it is usually accomplished by use of a multiplexer rather than by multiple A/D converters. The multiplexer sequentially connects the converter into the channel being sampled.

8.4.2 Digital-to-Analog (D/A) Converters

D/A converters, as mentioned in Section 8.1, are used to convert the digital words from the computer to a voltage level and are sometimes referred to as **Sample and Hold** devices. They provide analog outputs from a computer for driving actuators or perhaps a recording device such as an oscilloscope or strip-chart recorder. The basic idea behind their operation is that the binary bits cause switches (electronic gates) to open or close, thus routing the electric current through an appropriate network of resistors to generate the correct voltage level. Because no counting or iteration is required for such converters, they tend to be much faster than A/D converters. In fact, A/D converters that use the successive-approximation method of conversion include D/A converters as components.

8.4.3 Anti-Alias Prefilters

Analog prefilters reduce aliasing

An analog **anti-alias prefilter** is often placed between the sensor and the A/D converter. Its function is to reduce the higher-frequency noise components in the analog signal in order to prevent aliasing, that is, having the noise be modulated to a lower frequency by the sampling process.

An example of aliasing is shown in Fig. 8.14, where a 60 Hertz oscillatory signal is being sampled at 50 Hertz. The figure shows the result from the samples as a 10 Hertz signal and also shows the mechanism by which the frequency of the signal is aliased from 60 to 10 Hertz. Aliasing will occur any time the sample rate is not at least twice as fast as any of the frequencies in the signal being sampled. Therefore, to prevent aliasing of a 60 Hertz signal, the sample rate would have to be faster than 120 Hertz, clearly much higher than the 50 Hertz rate in the figure.

Nyquist–Shannon sampling theorem

Aliasing is one of the consequences of the **sampling theorem of Nyquist and Shannon**. Their theorem basically states that, for the signal to be accurately reconstructed from the samples, it must have no frequency component greater than half the sample rate ($^{\omega_s}/_2$). Another consequence of their theorem is that the highest frequency

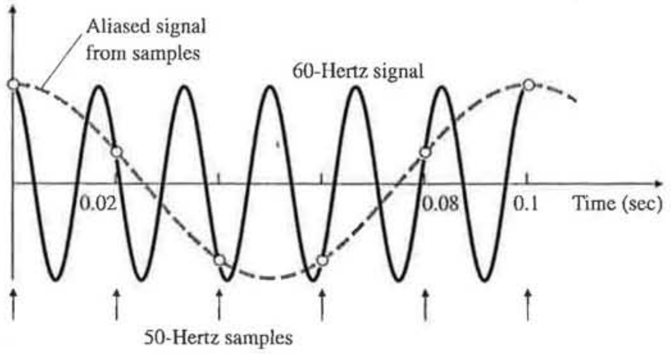


Figure 8.14
An example of aliasing

that can be unambiguously represented by discrete samples is the Nyquist rate of $\omega_s/2$, an idea we discussed in Section 8.2.3.

The consequence of aliasing on a digital control system can be substantial. In a continuous system, noise components with a frequency much higher than the control-system bandwidth normally have a small effect because the system will not respond at the high frequency. However, in a digital system, the frequency of the noise can be aliased down to the vicinity of the system bandwidth so that the closed-loop system would respond to the noise. Thus, the noise in a poorly designed digitally controlled system could have a substantially greater effect than if the control had been implemented using analog electronics.

The solution is to place an analog prefilter before the sampler. In many cases a simple first-order low-pass filter will do—that is,

$$H_p(s) = \frac{a}{s+a},$$

where the break point a is selected to be lower than $\omega_s/2$ so that any noise present with frequencies greater than $\omega_s/2$ is attenuated by the prefilter. The lower the break-point frequency selected, the more the noise above $\omega_s/2$ is attenuated. However, too low a break point may force the designer to reduce the control system's bandwidth. The prefilter does not completely eliminate the aliasing; however, through judicious choice of the prefilter break point and the sample rate, the designer has the ability to reduce the magnitude of the aliased noise to some acceptable level.

8.4.4 The Computer

The computer is the unit that does all the computations. Most digital controllers used today are built around a microcontroller that contains both a microprocessor and most of the other functions needed, including the A/D and D/A conversion. For development purposes in a laboratory, a digital controller could be a desktop-sized workstation or a PC. The relatively low cost of microprocessor technology has accounted for the large increase in the use of digital control systems, which started in the 1980s and continues into the 2000s.

The computer consists of a central processor unit (CPU), which does the computations and provides the system logic; a clock to synchronize the system; memory modules for data and instruction storage; and a power supply to provide the various required voltages. The memory modules come in three basic varieties:

1. **Read-only memory (ROM)** is the least expensive, but after its manufacture its contents cannot be changed. Most of the memory in products manufactured in quantity is ROM. It retains its stored values when power is removed.
2. **Random-access memory (RAM)** is the most expensive, but its values can be changed by the CPU. It is required only to store the values that will be changed during the control process and typically represents only a small fraction of the total memory of a developed product. It loses the values in memory when power is removed.
3. **Programmable read-only memory (EPROM)** is a ROM whose values can be changed by a technician using a special device. It is typically used during product development to enable the designer to try different algorithms and parameter values. It retains its stored values when power is removed. In some products, it is useful to have a few of the stored quantities in EPROMs so that individual calibrations can be carried out for each unit.

Microprocessors for control applications generally come with a digital word size of 8, 16, or 32 bits, although some have been available with 12 bits. Larger word sizes give better accuracy, but at an increase in cost. The most economical solution is often to use an 8-bit microprocessor, but to use two digital words to store one value (**double precision**) in the areas of the controller that are critical to the system accuracy. Many digital control systems use computers originally designed for digital signal-processing applications, so-called DSP chips.

8.5 Sample-Rate Selection

The selection of the best sample rate for a digital control system is the result of a compromise of many factors. Sampling too fast can cause a loss of accuracy while the basic motivation for lowering the sample rate ω_s is cost. A decrease in sample rate means more time is available for the control calculations; hence slower computers can be used for a given control function or more control capability can be achieved from a given computer. Either way, the cost per function is lowered. For systems with A/D converters, less demand on conversion speed will also lower cost. These economic arguments indicate that the best engineering choice is the slowest possible sample rate that still meets all performance specifications.

There are several factors that could provide a lower limit on the acceptable sample rate:

1. tracking effectiveness as measured by closed-loop bandwidth or by time-response requirements, such as rise time and settling time;
2. regulation effectiveness as measured by the error response to random plant disturbances;
3. error due to measurement noise and the associated prefilter design methods.

A fictitious limit occurs when using discrete equivalents. The inherent approximation in the method may give rise to decreased performance or even system instabilities as the sample rate is lowered. This can lead the designer to conclude that a faster sample rate is required. However, there are two solutions:

1. sample faster, and
2. recognize that the approximations are invalid and refine the design with a direct digital-design method described in the subsequent sections.

The ease of designing digital control systems with fast sample rates and the low cost of very capable computers often drives the designer to select a sample rate that is 40 × ω_{BW} or higher. For computers with fixed-point arithmetic, very fast sample rates can lead to multiplication errors that have the potential to produce significant offsets or limit cycles in the control (see Franklin et al., 1998).

8.5.1 Tracking Effectiveness

An absolute lower bound on the sample rate is set by a specification to track a command input with a certain frequency (the system bandwidth). The sampling theorem (see Section 8.4.3 and Franklin et al., 1998) states that in order to reconstruct an unknown, band-limited, continuous signal from samples of that signal, we must sample at least twice as fast as the highest frequency contained in the signal. Therefore, in order for a closed-loop system to track an input at a certain frequency, it must have a sample rate twice as fast; that is, ω_s must be at least twice the system bandwidth ($\omega_s \Rightarrow 2 \times \omega_{BW}$). We also saw from the results of mapping the s-plane into the z-plane ($z = e^{sT}$) that the highest frequency that can be represented by a discrete system is $\omega_s/2$, which supports the conclusion of the theorem.

It is important to note the distinction between the closed-loop bandwidth ω_{BW} and the highest frequency in the open-loop plant dynamics, because the two frequencies can be quite different. For example, closed-loop bandwidths can be an order of magnitude *less* than open-loop modes of resonances for some control problems. Information concerning the state of the plant resonances for purposes of control can be extracted from sampling the output without satisfying the sampling theorem because some a priori knowledge concerning these dynamics (albeit imprecise) is available, and the system is not required to track these frequencies. Thus a priori knowledge of the dynamic model of the plant can be included in the compensation in the form of a notch filter.

The closed-loop-bandwidth limitation provides the fundamental lower bound on the sample rate. In practice, however, the theoretical lower bound of sampling at twice the bandwidth of the reference input signal would not be judged sufficient in terms of the quality of the desired time responses. For a system with a rise time on the order of 1 sec (thus yielding a closed-loop bandwidth on the order of 0.5 Hertz), it is reasonable to insist on a sampling rate of 10 to 20 Hertz, which is a factor of 20 to 40 times ω_{BW}. The purposes of choosing a sample rate much greater than the bandwidth are to reduce the delay between a command and the system response to the command and also to smooth the system output to the control steps coming out of the ZOH.

8.5.2 Disturbance Rejection

Disturbance rejection is an important—if not the most important—aspect of any control system. Disturbances enter a system with various frequency characteristics ranging from steps to white noise. For the purpose of sample-rate selection, the higher-frequency random disturbances are the most influential.

The ability of the control system to reject disturbances with a good continuous controller represents the lower bound on the error response that we can hope for when implementing the controller digitally. In fact, some degradation relative to the continuous design must occur because the sampled values are slightly out of date at all times except precisely at the sampling instants. However, if the sample rate is very fast compared with the frequencies contained in the noisy disturbance, we should expect no appreciable loss from the digital system as compared with the continuous controller. At the other extreme, if the sample rate is very slow compared with the characteristic frequencies of the noise, the response of the system because of noise is essentially the same as the response we would get if the system had no control at all. The selection of a sample rate will place the response somewhere in between these two extremes. Thus, the impact of the sample rate on the ability of the system to reject disturbances may be very important to consider when choosing the sample rate.

Although the best choice of sample rate in terms of the ω_{BW} multiple is dependent on the frequency characteristics of the noise and the degree to which random disturbance rejection is important to the quality of the controller, sample rates on the order of 25 times ω_{BW} or higher are typical.

8.5.3 Effect of Anti-Alias Prefilter

Digital control systems with analog sensors typically include an analog anti-alias prefilter between the sensor and the sampler as described in Section 8.4.3. The prefilters are low-pass, and the simplest transfer function is

$$H_p(s) = \frac{a}{s+a},$$

so that the noise above the prefilter break point a is attenuated. The goal is to provide enough attenuation at half the sample rate ($^{\omega_s}/_2$) that the noise above $^{\omega_s}/_2$, when aliased into lower frequencies by the sampler, will not be detrimental to control system performance.

A conservative design procedure is to select ω_s and the break point to be sufficiently higher than the system bandwidth that the phase lag from the prefilter does not significantly alter the system stability. This would allow the prefilter to be ignored in the basic control system design. Furthermore, for a good reduction in the high-frequency noise at $^{\omega_s}/_2$, we choose a sample rate that is about 5 or 10 times higher than the prefilter break point. The implication of this prefilter design procedure is that sample rates need to be on the order of 30 to 100 times faster than the system bandwidth. Using this conservative design procedure, the prefilter influence will likely provide the lower bound on the selection of the sample rate.

An alternative design procedure is to allow significant phase lag from the prefilter at the system bandwidth. This requires us to include the analog prefilter characteristics in the plant model when carrying out the control design. It allows the use of lower sample rates, but at the possible expense of increased complexity in the compensation because additional phase lead must be provided to counteract the prefilter's phase lag. If this procedure is used and low prefilter break points are allowed, the effect of sample rate on sensor noise is small, and the prefilter essentially has no effect on the sample rate.

It may seem counterintuitive that placing a lag (the analog prefilter) in one portion of the controller and a counteracting lead [extra lead in $D(z)$] in another portion of the controller provides a net positive effect on the overall system. The net gain is a result of the fact that the lag is in the analog part of the system where high frequencies can exist. The counteracting lead is in the digital part of the system where frequencies above the Nyquist rate do not exist. The result is a reduction in the high frequencies before the sampling which are not reamplified by the counteracting digital lead, thus producing net reduction in high frequencies. Furthermore, these high frequencies are particularly insidious with a digital controller because of the aliasing that would result from the sampling.

8.5.4 Asynchronous Sampling

As noted in the previous paragraphs, divorcing the prefilter design from the control-law design may require using a faster sample rate than otherwise. This same result may show up in other types of architecture. For example, a smart sensor with its own computer running asynchronously relative to the primary control computer will not be amenable to direct digital design because the overall system transfer function depends on the phasing between the smart sensor and the primary digital controller. This situation is similar to that of the digitization errors discussed in Section 8.6. Therefore, if asynchronous digital subsystems are present, sample rates on the order of 20 × ω_{BW} or slower in any module should be used with caution and the system performance checked through simulation or experiment.

△ 8.6 Discrete Design

It is possible to obtain an exact discrete model that relates the samples of the continuous plant $y(k)$ to the input control sequence $u(k)$. This plant model can be used as part of a discrete model of the feedback system including the compensation $D(z)$. Analysis and design using this discrete model is called **discrete design** or, alternatively, **direct digital design**. The following subsections will describe how to find the discrete plant model (Section 8.6.1), what the feedback compensation looks like when designing with a discrete model (Sections 8.6.2 and 8.6.3), and how the design process is carried out (Section 8.6.4).

8.6.1 Analysis Tools

The exact discrete equivalent

The first step in performing a discrete analysis of a system with some discrete elements is to find the discrete transfer function of the continuous portion. For a system similar to that shown in Fig. 8.1(b), we wish to find the transfer function between $u(kT)$ and $y(kT)$. Unlike the cases discussed in the previous sections, there is an *exact* discrete equivalent for this system, because the ZOH precisely describes what happens between samples of $u(kT)$ and the output $y(kT)$ is dependent only on the input at the sample times $u(kT)$.

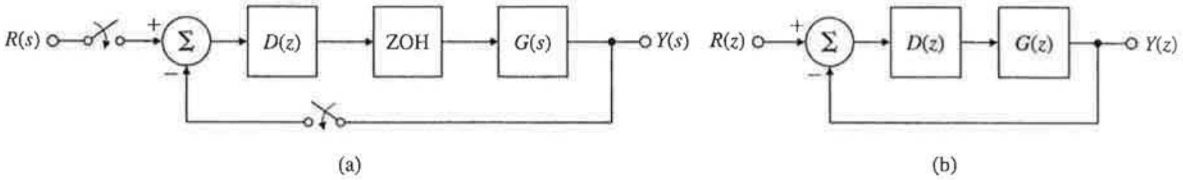


Figure 8.15

Comparison of (a) a mixed control system; and (b) its pure discrete equivalent

For a plant described by $G(s)$ and preceded by a ZOH, the discrete transfer function is

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}, \tag{8.33}$$

where $\mathcal{Z}\{F(s)\}$ is the z-transform of the sampled time series whose Laplace transform is the expression for $F(s)$, given on the same line in Table 8.1. Equation (8.33) has the term ${}^{G(s)}/_{s}$ because the control comes in as a step input from the ZOH during each sample period. The term $1 - z^{-1}$ reflects the fact that a one-sample duration step can be thought of as an infinite duration step followed by a negative step one cycle delayed. For a more complete derivation, see Franklin et al. (1998). Equation (8.33) allows us to replace the mixed (continuous and discrete) system shown in Fig. 8.15(a) with the equivalent pure discrete system shown in Fig. 8.15(b).

The analysis and design of discrete systems is very similar to the analysis and design of continuous systems; in fact, all the same rules apply. The closed-loop transfer function of Fig. 8.15(b) is obtained using the same rules of block-diagram reduction—that is,

$$\frac{Y(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)G(z)}. \tag{8.34}$$

To find the characteristic behavior of the closed-loop system, we need to find the factors in the denominator of Eq. (8.34)—that is, the roots of the discrete characteristic equation

$$1 + D(z)G(z) = 0.$$

The root-locus techniques used in continuous systems to find roots of a polynomial in s apply equally well and without modification to the polynomial in z; however, the interpretation of the results is quite different, as we saw in Fig. 8.4. A major difference is that the stability boundary is now the unit circle instead of the imaginary axis.

EXAMPLE 8.3 *Discrete Root Locus*

For the case in which $G(s)$ in Fig. 8.15(a) is

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

and $D(z) = K$, draw the root locus with respect to K, and compare your results with a root locus of a continuous version of the system. Discuss the implications of your loci.

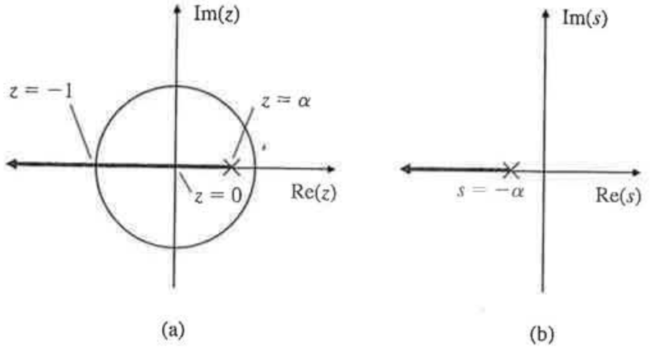


Figure 8.16
Root loci for (a) the z-plane; and (b) the s-plane

Solution. It follows from Eq. (8.33) that

$$\begin{aligned}G(z) &= (1-z^{-1})\mathcal{Z}\left[\frac{a}{s(s+a)}\right]\\ &= (1-z^{-1})\left[\frac{(1-e^{-aT})z^{-1}}{(1-z^{-1})(1-e^{-aT}z^{-1})}\right]\\ &= \frac{1-\alpha}{z-\alpha},\end{aligned}$$

where

$$\alpha = e^{-aT}.$$

To analyze the performance of the closed-loop system, standard root-locus rules apply. The result is shown in Fig. 8.16(a) for the discrete case and in Fig. 8.16(b) for the continuous case. In contrast to the continuous case, in which the system remains stable for all values of K, in the discrete case the system becomes oscillatory with decreasing damping ratio as z goes from 0 to -1 and eventually becomes unstable. This instability is due to the lagging effect of the ZOH, which is properly accounted for in the discrete analysis.

8.6.2 Feedback Properties

Discrete control laws

In continuous systems we typically start the design process by using the following basic design elements: proportional, derivative, or integral control laws, or some combination of these, sometimes with a lag included. The same ideas can be used in discrete design. Alternatively, the $D(z)$ resulting from the digitization of a continuously designed $D(s)$ will produce these basic design elements, which will then be used as a starting point in a discrete design. The discrete control laws are as follows:

Proportional

$$u(k) = Ke(k) \Rightarrow D(z) = K. \tag{8.35}$$

Derivative

$$u(k) = KT_D[e(k) - e(k-1)], \tag{8.36}$$

for which the transfer function is

$$D(z) = KT_D(1 - z^{-1}) = KT_D\frac{z-1}{z} = k_D\frac{z-1}{z}. \tag{8.37}$$

Integral

$$u(k) = u(k-1) + \frac{K_p}{T_I}e(k), \tag{8.38}$$

for which the transfer function is

$$D(z) = \frac{K}{T_I}\left(\frac{1}{1-z^{-1}}\right) = \frac{K}{T_I}\left(\frac{z}{z-1}\right) = k_I\left(\frac{z}{z-1}\right). \tag{8.39}$$

Lead Compensation

The examples in Section 8.3 showed that a continuous lead compensation leads to difference equations of the form

$$u(k+1) = \beta u(k) + K[e(k+1) - \alpha e(k)], \tag{8.40}$$

for which the transfer function is

$$D(z) = K\frac{1-\alpha z^{-1}}{1-\beta z^{-1}}. \tag{8.41}$$

8.6.3 Discrete Design Example

Digital control design consists of using the basic feedback elements of Eqs. (8.35) to (8.41) and iterating on the design parameters until all specifications are met.

EXAMPLE 8.4 *Direct Discrete Design of the Space Station Digital Controller*

Design a digital controller to meet the same specifications as in Example 8.2 using discrete design.

Solution. The discrete model of the $1/s^2$ plant, preceded by a ZOH, is found through Eq. (8.33) to be

$$G(z) = \frac{T^2}{2}\left[\frac{z+1}{(z-1)^2}\right],$$

which, with $T = 1$ sec, becomes

$$G(z) = \frac{1}{2}\left[\frac{z+1}{(z-1)^2}\right].$$

Proportional feedback in the continuous case yields pure oscillatory motion, so in the discrete case we should expect even worse results. The root locus in Fig. 8.17 verifies this. For very low values of K (where the locus represents roots at very low frequencies compared to the sample rate), the locus is tangent to the unit circle ($\zeta \cong 0$ indicating pure oscillatory motion), thus matching the proportional continuous design.

For higher values of K, Fig. 8.17 shows that the locus diverges into the unstable region because of the effect of the ZOH and sampling. To compensate for this, we will add a derivative term to the proportional term so that the control law is

$$U(z) = K[1 + T_D(1 - z^{-1})]E(z), \tag{8.42}$$

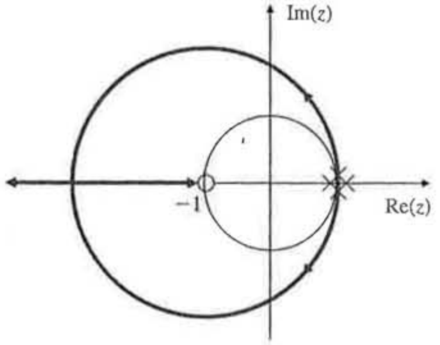


Figure 8.17
z-plane root locus for a $1/s^2$ plant with proportional feedback

which yields compensation of the form

$$D(z) = K\frac{z-\alpha}{z}, \tag{8.43}$$

where the new K and α replace the K and T_D in Eq. (8.42). Now the task is to find the values of α and K that yield good performance. The specifications for the design are that $\omega_n = 0.3$ rad/sec and $\zeta = 0.7$. Figure 8.4 indicates that this s-plane root location maps into a desired z-plane location of

$$z = 0.78 \pm 0.18j.$$

Figure 8.18 is the locus with respect to K for $\alpha = 0.85$. The location of the zero (at $z = 0.85$) was determined by trial and error until the locus passed through the desired z-plane location. The value of the gain when the locus passes through $z = 0.78 \pm 0.18j$ is $K = 0.374$. Equation (8.43) now becomes

$$D(z) = 0.374\frac{z-0.85}{z}. \tag{8.44}$$

Normally, it is not particularly advantageous to match specific z-plane root locations; rather it is necessary only to pick K and α (or T_D) to obtain acceptable z-plane roots, a much easier task. In this example, we want to match a specific location only so that we can compare the result with the design in Example 8.2.

The control law that results is

$$U(z) = 0.374(1 - 0.85z^{-1})E(z),$$

or

$$u(k) = 0.374e(k) - 0.318e(k-1), \tag{8.45}$$

which is similar to the control equation (8.29) obtained previously.

The controller in Eq. (8.45) basically differs from the continuously designed controller [Eq. (8.29)] only in the absence of the $u(k-1)$ term. The $u(k-1)$ term in Eq. (8.29) results from the lag term $(s + b)$ in the compensation [Eq. (8.27)]. The lag term is typically included in analog controllers both because it supplies noise attenuation and because pure analog differentiators are difficult to build. Some equivalent lag in discrete design naturally appears as a pole at $z = 0$ (see Fig. 8.18) and represents the one-sample delay in computing the derivative by a first difference. For more noise attenuation, we could move the pole to the right of $z = 0$, thus resulting in less derivative action and more smoothing, the same trade-off that exists in continuous control design.

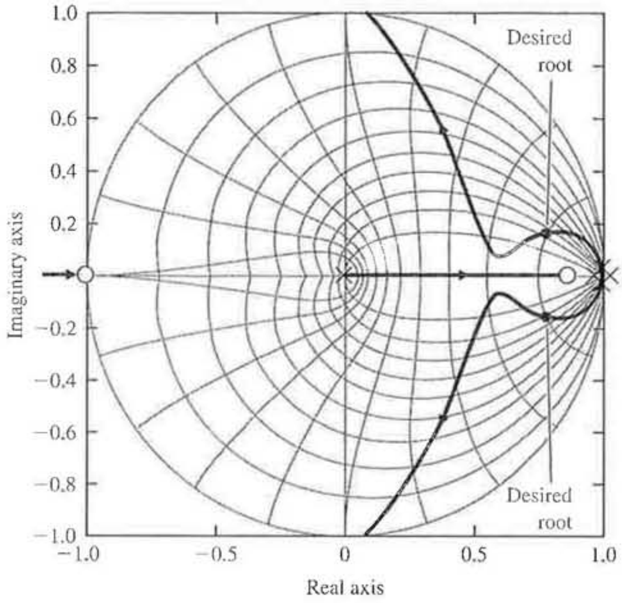


Figure 8.18
z-plane locus for the $^1/_{s^2}$ plant with $D(z) = K(z - 0.85)/z$

8.6.4 Discrete Analysis of Designs

Any digital controller, whether designed by discrete equivalents or directly in the z-plane, can be analyzed using discrete analysis, which consists of the following steps:

1. Find the discrete model of the plant and ZOH using Eq. (8.33).
2. Form the feedback system including $D(z)$.
3. Analyze the resulting discrete system.

We can determine the roots of the system using a root locus, as described in Section 8.6.3, or we can determine the time history (at the sample instants) of the discrete system.

EXAMPLE 8.5 *Damping and Step Response in Digital versus Continuous Design*

Use discrete analysis to determine the equivalent s-plane damping and the step responses of the digital designs in Examples 8.2 and 8.4, and compare your results with the damping and step response of the continuous case in Example 8.2.

Solution. The MATLAB statements to evaluate the damping and step response of the continuous case in Example 8.2 are

```
sysGs = tf(1,[1  0  0]);
sysDs = tf(0.81 * [1  0.2],[1  2]);
sysGDs = series(sysGs,sysDs);
sysCLs = feedback(sysGDs,1,1);
step(sysCLs)
damp(sysCLs)
```

To analyze the digital control cases, the model of the plant preceded by the ZOH is found using the statements

```
T = 1;
sysGz = c2d(sysGs,T,'zoh')
```

Analysis of the digital control designed using the discrete equivalent [Eq. (8.29)] in Example 8.2 is performed by the statements

```
sysDz = tf( [.389 –.319],[1 –.135])
sysDGz = series(sysGz,sysDz)
sysCLz = feedback(sysDGz,1)
step(sysCLz,T)
damp(sysCLz,T)
```

Likewise, the discrete design of $D(z)$ from Eq. (8.44) can be analyzed by the same sequence.

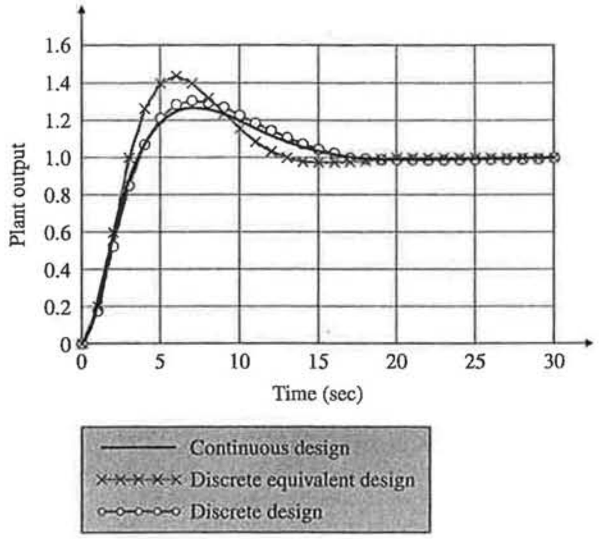
The resulting step responses are shown in Fig. 8.19. The calculated damping ζ and complex root natural frequencies ω_n of the closed-loop systems are

$$\begin{aligned} \text{Continuous case:} &\quad \zeta = 0.705, \quad \omega_n = 0.324; \\ \text{Discrete equivalent:} &\quad \zeta = 0.645, \quad \omega_n = 0.441; \\ \text{Discrete design:} &\quad \zeta = 0.733, \quad \omega_n = 0.306. \end{aligned}$$

The figure shows increased overshoot for the discrete equivalent method that occurred because of the decreased damping of that case. Very little increased overshoot occurred in the discrete design, because that compensation was adjusted specifically so that the equivalent s-plane damping of the discrete system was approximately at the desired damping value of $\zeta = 0.7$.

Figure 8.19

Step response of the continuous and digital systems in Examples 8.2 and 8.4



Although the analysis showed some differences between the performance of the digital controllers designed by the two methods, neither the performance nor the control equations [Eqs. (8.29) and (8.45)] are very different. This similarity results because the sample rate is fairly fast compared with ω_n—that is, $\omega_s \cong 20 \times \omega_n$. If we were to decrease the sample rate, the numerical values in the compensations would become increasingly different and the performance would degrade considerably for the discrete equivalent case.

As a general rule, discrete design should be used if the sampling frequency is slower than $10 \times \omega_n$. At the very least, a discrete equivalent design with slow sampling ($\omega_s < 10 \times \omega_n$) should be verified by a discrete analysis or by simulation, as described in Section 4.4, and the compensation adjusted if needed. A simulation of a digital control system is a good idea in any case. If it properly accounts for all delays and possibly asynchronous behavior of different modules, it may expose instabilities that are impossible to detect using continuous or discrete linear analysis. A more complete discussion regarding the effects of sample rate on the design is contained in Section 8.5.

8.7 Historical Perspective

One of the earliest examples of actual control of systems based on sampled data came with the use of search RADAR in WWII. In that case, the position of the target was available only once each revolution of the antenna. The theory of sampled data systems was developed by the mathematician W. Hurewicz[2] and published as a chapter in H. M. James, N. B. Nichols, and R. S. Phillips, *Theory of Servomechanisms*, vol. 25, Rad Lab Series, New York, McGraw Hill, 1947. The historical perspective for Chapter 5 discussed the introduction of computers for engineers performing design activities. The possibility of using computers for direct digital control motivated the continuation of work on sampled data systems during the 1950s, especially at Columbia University under Professor J. R. Ragazzini. That work was published in J. R. Ragazzini and G. F. Franklin, *Sampled-Data Control Systems*, New York, McGraw Hill, 1958. Early applications were in the process control industry where the relatively large and expensive computers available at the time could be justified. Professor Karl Astrom introduced direct digital control of a paper mill in Sweden in the early 1960s.

In 1961, when President Kennedy announced the goal of sending a man to the moon, there were no digital autopilots for aerospace vehicles. In fact, small digital computers suitable for implementing control systems were virtually nonexistent. The team at the MIT Draper Labs (called the Instrumentation Lab at that time) in charge of designing and building the Apollo control systems initially designed the control systems for the lunar and command modules with conventional analog electronics. However, they discovered that those systems would be too heavy and complex for the mission. So the decision was made to design and build the first aerospace digital

[2] Hurewicz died in 1956 falling off a ziggurat (a Mexican pyramid) on a conference outing at the International Symposium on algebraic topology in Mexico. It is suggested that he was: "... a paragon of absentmindedness, a failing that probably led to his death."

control system. Bill Widnall, Dick Battin, and Don Fraser were all key players in the successful design and execution of that system for the Apollo flights in late 1960s. The group went on to demonstrate a digital autopilot for NASA's F-8 in the 1970s, and digital autopilots went on to become dominant over the 1980s and beyond. In fact, with the introduction of inexpensive digital signal processors, most control systems of any kind became digital by the turn of the century and, today, very few control systems are being implemented with analog electronics. This evolution has had an effect on the training for controls engineers. In the past, the ability to design and build the specialized circuitry for analog electronic controls caused many controls engineers to have an Electrical Engineering background. Now, with easily programmable digital computers being readily available, the background of controls engineers tends more toward the specialties that are most familiar with the systems being controlled.

SUMMARY

- The simplest and most expedient design technique is to transform a continuous controller design to its discrete form—that is, to use its **discrete equivalent**.
- **Design using discrete equivalents** entails (a) finding the continuous compensation $D(s)$ using the ideas in Chapters 1 to 7, and (b) approximating $D(s)$ with difference equations using Tustin's method or the matched-pole–zero method.
- In order to analyze a discrete controller design, or any discrete system, the z-**transform** is used to determine the system's behavior. The z-transform of a time sequence $f(k)$ is given by

  $$\mathcal{Z}\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}$$

  and has the key property that

  $$\mathcal{Z}\{f(k-1)\} = z^{-1}F(z).$$

  This property allows us to find the discrete transfer function of a **difference equation**, which is the digital equivalent of a differential equation for continuous systems. Analysis using z-transforms closely parallels that using Laplace transforms.
- Normally z-transforms are found using the computer (MATLAB) or looking up in Table 8.1.
- The discrete Final Value Theorem is

  $$\lim_{k\to\infty} x(k) = \lim_{z\to 1}(1 - z^{-1})X(z),$$

  provided that all poles of $(1 - z^{-1})X(z)$ are inside the unit circle.
- For a continuous signal $f(t)$ whose samples are $f(k)$, the poles of $F(s)$ are related to the poles of $F(z)$ by

  $$z = e^{sT}.$$

- The following are the most common discrete equivalents:
  1. **Tustin's approximation**:
     $$D(z) = D(s)|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$
  2. the **matched pole–zero approximation:**
     - Map poles and zeros by $z = e^{sT}$.
     - Add powers of $z + 1$ to the numerator until numerator and denominator are of equal order or the numerator is one order less than the denominator.
     - Set the low-frequency gain of $D(z)$ equal to that of $D(s)$.
- If designing by discrete equivalents, a minimum **sample rate** of 20 times the bandwidth is recommended. Typically, even faster sampling is useful for best performance.
- Analog **prefilters** are commonly placed before the **sampler** in order to attenuate the effects of high-frequency measurement noise. A sampler **aliases** all frequencies in the signal that are greater than half the sample frequency to lower frequencies; therefore, prefilter break points should be selected so that no significant frequency content remains above half the sample rate.
- The discrete model of the continuous plant $G(s)$ preceded by a **ZOH** is
  $$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}.$$
  The discrete plant model plus the discrete controller can be analyzed using the **z-transform** or simulated using SIMULINK.
- **Discrete design** is an exact design method and avoids the approximations inherent with discrete equivalents. The design procedure entails (a) finding the discrete model of the plant $G(s)$, and (b) using the discrete model to design the compensation directly in its discrete form. The design process is more cumbersome than discrete equivalent design and requires that a sample rate be selected before commencing the design. A practical approach is to commence the design using discrete equivalents, then tune up the result using discrete design.
- Discrete design using $G(z)$ closely parallels continuous design, but the stability boundary and interpretation of z-plane root locations are different. Figure 8.5 summarizes the response characteristics.
- If using discrete design, system stability can theoretically be assured when **sampling** at a rate as slow as twice the bandwidth. However, for good transient performance and random disturbance rejection, best results are obtained by sampling at 10 times the closed-loop bandwidth or faster. In some cases with troublesome vibratory modes, it is sometimes useful to sample more than twice as fast as the vibratory mode.

REVIEW QUESTIONS

1. What is the Nyquist rate? What are its characteristics?
2. Describe the discrete equivalent design process.
3. Describe how to arrive at a $D(z)$ if the sample rate is $30 \times \omega_{BW}$.

4. For a system with a 1 rad/sec bandwidth, describe the consequences of various sample rates.
5. Give two advantages for selecting a digital processor rather than analog circuitry to implement a controller.
6. Give two disadvantages for selecting a digital processor rather than analog circuitry to implement a controller.

△ 7. Describe how to arrive at a $D(z)$ if the sample rate is $5 \times \omega_{BW}$.

PROBLEMS

Problems for Section 8.2: Dynamic Analysis of Discrete Systems

8.1 The z-transform of a discrete-time filter $h(k)$ at a 1 Hertz sample rate is

$$H(z) = \frac{1 + (1/2)z^{-1}}{[1 - (1/2)z^{-1}][1 + (1/3)z^{-1}]}.$$

(a) Let $u(k)$ and $y(k)$ be the discrete input and output of this filter. Find a difference equation relating $u(k)$ and $y(k)$.

(b) Find the natural frequency and damping coefficient of the filter's poles.

(c) Is the filter stable?

8.2 Use the z-transform to solve the difference equation

$$y(k) - 3y(k-1) + 2y(k-2) = 2u(k-1) - 2u(k-2),$$

where

$$u(k) = \begin{cases} k, & k \geq 0, \\ 0, & k < 0, \end{cases}$$

$$y(k) = 0, \qquad k < 0.$$

8.3 The one-sided z-transform is defined as

$$F(z) = \sum_{0}^{\infty} f(k)z^{-k}.$$

(a) Show that the one-sided transform of $f(k+1)$ is $\mathcal{Z}\{f(k+1)\} = zF(z) - zf(0)$.

(b) Use the one-sided transform to solve for the transforms of the Fibonacci numbers generated by the difference equation $u(k+2) = u(k+1) + u(k)$. Let $u(0) = u(1) = 1$. [*Hint:* You will need to find a general expression for the transform of $f(k+2)$ in terms of the transform of $f(k)$.]

(c) Compute the pole locations of the transform of the Fibonacci numbers.

(d) Compute the inverse transform of the Fibonacci numbers.

(e) Show that, if $u(k)$ represents the kth Fibonacci number, then the ratio $u(k+1)/u(k)$ will approach $(\frac{1+\sqrt{5})}{2}$. This is the golden ratio valued so highly by the Greeks.

8.4 Prove the seven properties of the s-plane-to-z-plane mapping listed in Section 8.2.3.