

CHAPTER 2

Linear, Discrete, Dynamic-Systems Analysis: The z-Transform

2.1 INTRODUCTION

The unique element in the structure of Fig. 1.1 is the digital computer. The fundamental character of the digital computer is that it takes a finite time to compute answers, and it does so with only finite precision. The purpose of this chapter is to develop tools of analysis necessary to understand and to guide the design of programs for a computer acting as a linear, dynamic control component. Needless to say, digital computers can do many things other than control linear, dynamic systems; it is our purpose in this chapter to examine their characteristics when doing this elementary control task and to develop the basic analysis tools needed to write programs for a real-time control computer.

2.2 LINEAR DIFFERENCE EQUATIONS

We assume that the analog-to-digital converter (A/D) in Fig. 1.1 takes samples of the signal y at discrete times and passes them to the computer so that $\hat{y}(kT) = y(kT)$. The job of the computer is to take these sample values and compute in some fashion the signals to be put out through the digital-to-analog converter (D/A). The characteristics of the A/D and D/A converters will be discussed later. Here we consider the treatment of the data inside the computer. Suppose we call the input signals up to the k th sample $e_0, e_1, e_2, \dots, e_k$, and the output signals prior to that time $u_0, u_1, u_2, \dots, u_{k-1}$. Then, to get the next output, we have the machine com-

pute some function, which we can express in symbolic form as

$$u_k = f(e_0, \dots, e_k; u_0, \dots, u_{k-1}). \quad (2.1)$$

Because we plan to emphasize the elementary and the dynamic possibilities, we assume that the function f in (2.1) is *linear* and depends on only a *finite* number of past e 's and u 's. Thus we write

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} - \dots - a_n u_{k-n} + b_0 e_k + b_1 e_{k-1} + \dots + b_m e_{k-m}. \quad (2.2)$$

Equation (2.2) is called a linear recurrence equation or difference equation and, as we shall see, has many similarities with a linear differential equation. The name "difference equation" derives from the fact that we could write (2.2) using u_k plus the differences in u_k , which are defined as

$$\begin{aligned} \nabla u_k &= u_k - u_{k-1} && \text{(first difference),} \\ \nabla^2 u_k &= \nabla u_k - \nabla u_{k-1} && \text{(second difference),} \\ \nabla^n u_k &= \nabla^{n-1} u_k - \nabla^{n-1} u_{k-1} && \text{(} n\text{th difference).} \end{aligned} \quad (2.3)$$

If we solve (2.3) for the values of u_k , u_{k-1} , and u_{k-2} in terms of differences, we find

$$\begin{aligned} u_k &= u_k, \\ u_{k-1} &= u_k - \nabla u_k, \\ u_{k-2} &= u_k - 2\nabla u_k + \nabla^2 u_k. \end{aligned}$$

Thus, for a second-order equation with coefficients a_1 , a_2 , and b_0 (we let $b_1 = b_2 = 0$ for simplicity), we find the equivalent difference equation to be

$$a_2 \nabla^2 u_k - (a_1 + 2a_2) \nabla u_k + (a_2 + a_1 + 1) u_k = b_0 e_k.$$

Although the two forms are equivalent, the recurrence form of (2.2) is more convenient for computer implementation; we will drop the form using differences. We will continue, however, to refer to our equations as "difference equations." If the a 's and b 's in (2.2) are constant, then the computer is solving a *constant-coefficient difference equation* (CCDE). We plan to demonstrate later that with such equations the computer can control linear constant dynamic systems and approximate most of the other tasks of linear, constant, dynamic systems, including performing the functions of electronic filters. To do so, it is necessary first to examine methods of obtaining solutions to (2.2) and to study the general properties of these solutions.

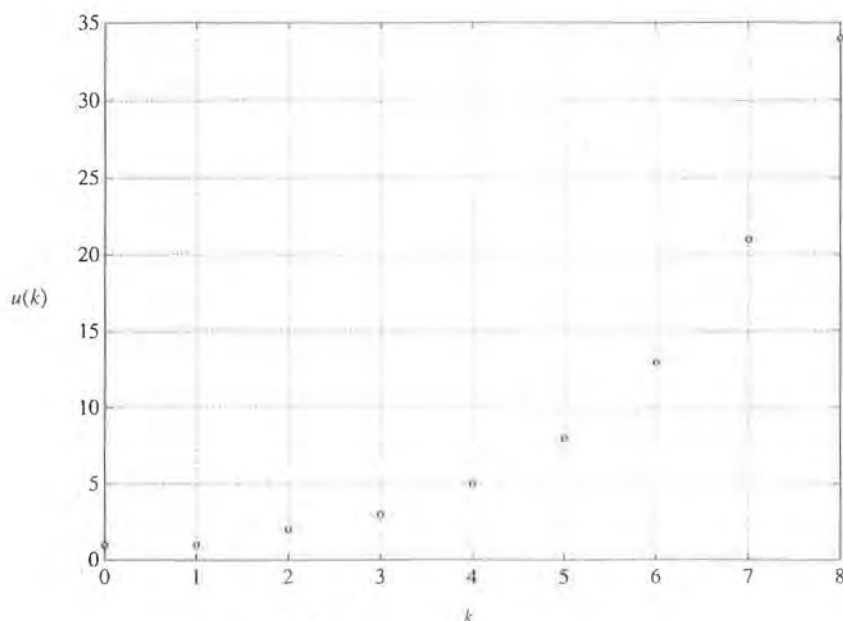


Figure 2.1 The Fibonacci numbers.

To solve a specific CCDE is an elementary matter. We need a starting time (k -value) and some initial conditions to characterize the contents of the computer memory at this time. For example, suppose we take the case

$$u_k = u_{k-1} + u_{k-2} \quad (2.4)$$

and start at $k = 2$. Here there are no input values, and to compute u_2 we need to know the (initial) values for u_0 and u_1 . Let us take them to be $u_0 = u_1 = 1$. The first nine values are 1, 1, 2, 3, 5, 8, 13, 21, 34.... A plot of the values of u_k versus k is shown in Fig. 2.1.

The results, the Fibonacci numbers, are named after the thirteenth-century mathematician¹ who studied them. For example, (2.4) has been used to model the growth of rabbits in a protected environment.² However that may be, the output of the system represented by (2.4) would seem to be

¹Leonardo Fibonacci of Pisa, who introduced Arabic notation to the Latin world about 1200 A.D.

²Wilde (1964). Assume that u_k represents pairs of rabbits and that babies are born in pairs. Assume that no rabbits die and that a new pair begin reproduction after one period. Thus at time k , we have all the old rabbits, u_{k-1} , plus the newborn pairs born to the mature rabbits, which are u_{k-2} .

growing, to say the least. If the response of a dynamic system to any finite initial conditions can grow without bound, we call the system *unstable*. We would like to be able to examine equations like (2.2) and, without having to solve them explicitly, see if they are stable or unstable and even understand the general shape of the solution.

One approach to solving this problem is to assume a form for the solution with unknown constants and to solve for the constants to match the given initial conditions. For continuous, ordinary, differential equations that are constant and linear, exponential solutions of the form e^{st} are used. In the case of linear, constant, difference equations, it turns out that solutions of the form z^k will do where z has the role of s and k is the discrete independent variable replacing time, t . Consider (2.4). If we assume that $u(k) = Az^k$, we get the equation

$$Az^k = Az^{k-1} + Az^{k-2}.$$

Now if we assume $z \neq 0$ and $A \neq 0$, we can divide by A and multiply by z^{-k} , with the result

$$1 = z^{-1} + z^{-2}$$

or

$$z^2 = z + 1.$$

This polynomial of second degree has two solutions, $z = 1/2 \pm \sqrt{5}/2$. Let's call these z_1 and z_2 . Since our equation is linear, a sum of the individual solutions will also be a solution. Thus, we have found that a solution to (2.4) is of the form

$$u(k) = A_1 z_1^k + A_2 z_2^k.$$

We can solve for the unknown constants by requiring that this general solution satisfy the specific initial conditions given. If we substitute $k = 0$ and $k = 1$, we obtain the simultaneous equations

$$1 = A_1 + A_2,$$

$$1 = A_1 z_1 + A_2 z_2.$$

These equations are easily solved³ to give

$$A_1 = \frac{1 + \sqrt{5}}{2\sqrt{5}},$$

$$A_2 = \frac{\sqrt{5} - 1}{2\sqrt{5}}.$$

And now we have the complete solution of (2.4) in a closed form. Furthermore, we can see that since $z_1 = (1 + \sqrt{5})/2$ is greater than 1, the term in z_1^k will grow without bound as k grows, which confirms our suspicion that the equation represents an unstable system. We can generalize this result. The equation in z that we obtain after we substitute $u = z^k$ is a polynomial in z known as the *characteristic equation* of the difference equation. If any solution of this equation is outside the unit circle (has a magnitude greater than one), the corresponding difference equation is unstable in the specific sense that for some finite initial conditions the solution will grow without bound as time goes to infinity. If *all* the roots of the characteristic equation are *inside* the unit circle, the corresponding difference equation is stable.

Example 2.1: Is the equation

$$u(k) = 0.9u(k-1) - 0.2u(k-2)$$

stable? The characteristic equation is

$$z^2 - 0.9z + 0.2 = 0,$$

and the characteristic roots are $z = 0.5$ and $z = 0.4$. Since both these roots are inside the unit circle, the equation is stable.

As an example of the origins of a difference equation with an external input, we consider the discrete approximation to integration. Suppose we have a continuous signal, $e(t)$, of which a segment is sketched in Fig. 2.2, and we wish to compute an approximation to the integral

$$\mathcal{J} = \int_0^t e(t) dt, \quad (2.5)$$

³Do it!

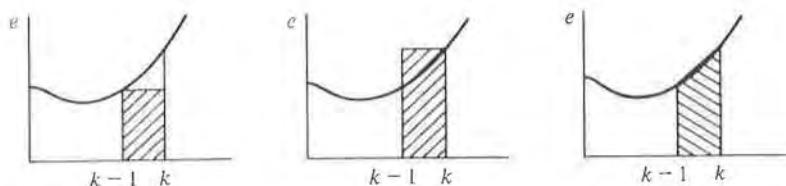


Figure 2.2 Plot of a function and alternative approximations to the area under the curve over a single time interval.

using only the discrete values $e(0), \dots, e(t_{k-1}), e(t_k)$. We assume that we have an approximation for the integral from zero to the time t_{k-1} and we call it u_{k-1} . The problem is to obtain u_k from this information. Taking the view of the integral as the area under the curve $e(t)$, we see that this problem reduces to finding an approximation to the area under the curve between t_{k-1} and t_k . Three alternatives are sketched in Fig. 2.2. We can use the rectangle of height e_{k-1} , or the rectangle of height e_k , or the trapezoid formed by connecting e_{k-1} to e_k by a straight line. If we take the third choice, the area of the trapezoid is

$$A = \frac{t_k - t_{k-1}}{2} (e_k + e_{k-1}). \quad (2.6)$$

Finally, if we assume that the sampling period, $t_k - t_{k-1}$, is a constant, T , we are led to a simple formula for discrete (trapezoid rule) integration:

$$u_k = u_{k-1} + \frac{T}{2} (e_k + e_{k-1}). \quad (2.7)$$

If $e(t) = t$, then $e_k = kT$ and substitution of $u_k = (T^2/2)k^2$ satisfies (2.7) and is exactly the integral of e . [It should be, because if $e(t)$ is a straight line, the trapezoid is the *exact* area.] If we approximate the area under the curve by the rectangle of height e_{k-1} , the result is called the Forward Rectangular Rule and is described by

$$u_k = u_{k-1} + Te_{k-1}.$$

A third possibility is the Backward Rectangular Rule, given by

$$u_k = u_{k-1} + Te_k.$$

Each of these integration rules is a special case of our general difference equation (2.2). We will examine the properties of these rules later, in Chapter 4, while discussing means to obtain a difference equation that will be equivalent to a given differential equation.

Thus we see that difference equations can be evaluated directly by a digital computer and that they can represent models of physical processes and approximations to integration. It turns out that if the difference equations are linear with coefficients that are constant, we can describe the relation between u and e by a transfer function, and thereby gain a great aid to analysis and also to the design of linear, constant, discrete controls.

2.3 THE DISCRETE TRANSFER FUNCTION

We will obtain the transfer function of linear, constant, discrete systems by the method of z -transform analysis. A logical alternative viewpoint that requires a bit more mathematics but has some appeal is given in Section 2.7.2. The results are the same. We also show how these same results can be expressed in the state space form in Section 2.3.3.

2.3.1 The z -Transform

If a signal has discrete values $e_0, e_1, \dots, e_k, \dots$ we define the z -transform of the signal as the function^{4,5}

$$\begin{aligned} E(z) &\triangleq \mathcal{Z}\{e_k\} \\ &\triangleq \sum_{k=-\infty}^{\infty} e_k z^{-k}, \quad r_0 < |z| < R_0, \end{aligned} \quad (2.8)$$

⁴We use the notation \triangleq to mean "is defined as."

⁵In (2.8) the lower limit is $-\infty$ so that values of e_k on both sides of $k = 0$ are included. The transform so defined is sometimes called the two-sided z -transform to distinguish it from the one-sided definition, which would be $\sum_0^{\infty} e_k z^{-k}$. For signals that are zero for $k < 0$, the transforms obviously give identical results. To take the one-sided transform of u_{k-1} , however, we must handle the value of u_{-1} , and thus are initial conditions introduced by the one-sided transform. Examination of this property and other features of the one-sided transform are invited by the problems. We select the two-sided transform because we need to consider signals that extend into negative time when we study random signals in Chapter 8.

and we assume we can find values of r_0 and R_0 as bounds on the magnitude of the complex variable z for which the series (2.8) converges. A discussion of convergence is deferred until Section 2.7.

Example 2.2: As an example to illustrate (2.8), consider that the data e_k are taken as samples from the time signal $e^{-at}1(t)$ at sampling period T where $1(t)$ is the unit step function, zero for negative t , and one for positive t . Then $e_k = e^{-akT}1(kT)$. The z -transform of this is

$$\begin{aligned}\sum_{k=-\infty}^{\infty} e_k z^{-k} &= \sum_0^{\infty} e^{-akT} z^{-k} \\ &= \sum_0^{\infty} (e^{-aT} z^{-1})^k \\ &= \frac{1}{1 - e^{-aT} z^{-1}} \quad e^{-aT} < |z| < \infty \\ &= \frac{z}{z - e^{-aT}} \quad |z| > e^{-aT}.\end{aligned}$$

We will return to the analysis of signals and development of a table of useful z -transforms in Section 2.5; we first examine the use of the transform to reduce difference equations to algebraic equations and techniques for representing these as block diagrams.

2.3.2 The Transfer Function

The z -transform has the same role in discrete systems that the Laplace transform has in analysis of continuous systems. For example, the z -transforms for e_k and u_k in the difference equation (2.2) or in the trapezoid integration (2.7) are related in a simple way that permits the rapid solution of linear, constant, difference equations of this kind. To find the relation, we proceed by direct substitution. We take the definition given by (2.8) and, in the same way, we define the z -transform of the sequence $\{u_k\}$ as

$$U(z) \triangleq \sum_{k=-\infty}^{\infty} u_k z^{-k}. \quad (2.9)$$

Now we multiply (2.7) by z^{-k} and sum over k . We get

$$\sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} + \frac{T}{2} \left(\sum_{k=-\infty}^{\infty} e_k z^{-k} + \sum_{k=-\infty}^{\infty} e_{k-1} z^{-k} \right). \quad (2.10)$$

From (2.9), we recognize the left-hand side as $U(z)$. In the first term on the right, we let $k-1 = j$ to obtain

$$\sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} = \sum_{j=-\infty}^{\infty} u_j z^{-(j+1)} = z^{-1} U(z). \quad (2.11)$$

By similar operations on the third and fourth terms we can reduce (2.10) to

$$U(z) = z^{-1} U(z) + \frac{T}{2} [E(z) + z^{-1} E(z)]. \quad (2.12)$$

Equation (2.12) is now simply an algebraic equation in z and the functions U and E . Solving it we obtain

$$U(z) = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} E(z). \quad (2.13)$$

We define the ratio of the transform of the output to the transform of the input as the *transfer function*, $H(z)$. Thus, in this case, the transfer function for trapezoid-rule integration is

$$\frac{U(z)}{E(z)} \triangleq H(z) = \frac{T}{2} \frac{z + 1}{z - 1}. \quad (2.14)$$

For the more general relation given by (2.2), it is readily verified by the same techniques that

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}},$$

and if $n \geq m$, we can write this as a ratio of polynomials in z as

$$\begin{aligned} H(z) &= \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_n} \\ &= \frac{b(z)}{a(z)}. \end{aligned} \quad (2.15)$$

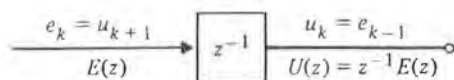


Figure 2.3 The unit delay.

The general input-output relation between transforms with linear, constant, difference equations is

$$U(z) = H(z)E(z). \quad (2.16)$$

Although we have developed the transfer function with the z -transform, it is also true that the transfer function is the ratio of the output to the input when both vary as z^k .

Because $H(z)$ is a rational function of a complex variable, we use the terminology of that subject. Suppose we call the numerator polynomial $b(z)$ and the denominator $a(z)$. The places in z where $b(z) = 0$ are *zeros* of the transfer function, and the places in z where $a(z) = 0$ are the *poles* of $H(z)$. If z_0 is a pole and $(z - z_0)^p H(z)$ has neither pole nor zero at z_0 , we say that $H(z)$ has a pole of order p at z_0 . If $p = 1$, the pole is simple. The transfer function (2.14) has a simple pole at $z = 1$ and a simple zero at $z = -1$.

We can now give a physical meaning to the variable z . Suppose we let all coefficients in (2.15) be zero except b_1 and we take b_1 to be 1. Then $H(z) = z^{-1}$. But $H(z)$ represents the transform of (2.2), and with these coefficient values the difference equation reduces to

$$u_k = e_{k-1}. \quad (2.17)$$

The present value of the output, u_k , equals the input *delayed by one period*. Thus we see that a transfer function of z^{-1} is a *delay* of one time unit. We can picture the situation as in Fig. 2.3, where both time and transform relations are shown.

Since the relations of (2.7), (2.14), and (2.15) are all composed of delays, they can be expressed in terms of z^{-1} ! Consider (2.7). In Fig. 2.4 we illustrate the difference equation (2.7) using the transfer function z^{-1} as the symbol for a unit delay.

We can follow the operations of the discrete integrator by tracing the signals through Fig. 2.4. For example, the present value of e_k is passed to the first summer, where it is added to the previous value e_{k-1} , and the sum is multiplied by $T/2$ to compute the area of the trapezoid between e_{k-1} and e_k . This is the signal marked a_k in Fig. 2.4. After this, there is another sum, where the previous output, u_{k-1} , is added to the new area to form the next

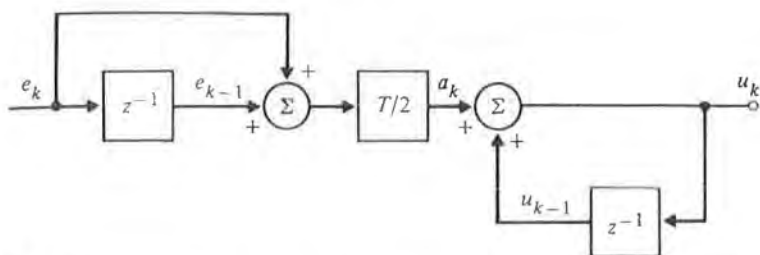


Figure 2.4 A block diagram of trapezoid integration as represented by (2.7).

value of the integral estimate, u_k . The discrete integration occurs in the loop with one delay, z^{-1} , and unity gain.

2.3.3 Block Diagrams and State-Variable Descriptions

Because (2.16) is a linear algebraic relationship, a system of such relations is described by a system of linear equations. These can be solved by the methods of linear algebra or by the graphical methods of block diagrams. To use block-diagram analysis to manipulate these discrete-transfer-function relationships, there are only four primitive cases:

1. The transfer function of paths in parallel is the sum of the single-path transfer functions (Fig. 2.5).
2. The transfer function of paths in series is the *product* of the path transfer functions (Fig. 2.6).
3. The transfer function of a single loop of paths is the transfer function of the forward path divided by one minus the loop transfer function (Fig. 2.7).
4. The transfer function of an arbitrary multipath diagram is given by combinations of these cases. Mason's rule⁶ can also be used.

For the general difference equation of (2.2), we already have the transfer function in (2.15). It is interesting to connect this case with a block diagram *using only simple delay forms for z* in order to see several "canonical" block diagrams and to introduce the description of discrete systems using equations of state.

There are many ways to reduce the difference equation (2.2) to a block diagram involving z only as the delay operator, z^{-1} . The first one we will

⁶Mason (1956). See Franklin, Powell, and Emami-Naeini (1986) for a discussion.

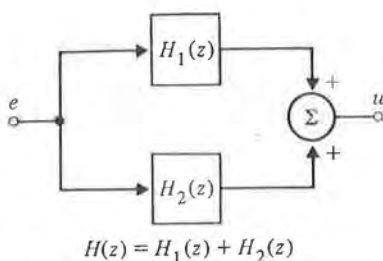


Figure 2.5 Block diagram of parallel blocks.

consider leads to the “control” canonical form. We begin with the transfer function as a ratio of polynomials

$$U(z) = H(z)E(z) = \frac{b(z)}{a(z)}E(z) = b(z)\xi,$$

where

$$\xi = \frac{E(z)}{a(z)}$$

and thus

$$a(z)\xi = E(z).$$

At this point we need to get specific; and rather than carry through with a system of arbitrary order, we will work out the details for the third-order case and leave it to the reader to extend the results in the obvious way to whatever order is desired. In the development that follows, we will consider the variables u , e , and ξ as *time* variables and z as an advance operator such that $zu(k) = u(k+1)$ or $z^{-1}u = u(k-1)$. With this convention (which is simply using the property of z derived earlier), consider the equations

$$(z^3 + a_1z^2 + a_2z + a_3)\xi = e, \quad (2.18)$$

$$(b_0z^3 + b_1z^2 + b_2z + b_3)\xi = u. \quad (2.19)$$

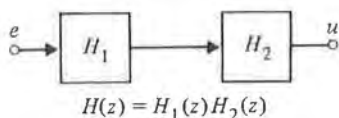


Figure 2.6 Block diagram of cascade blocks.

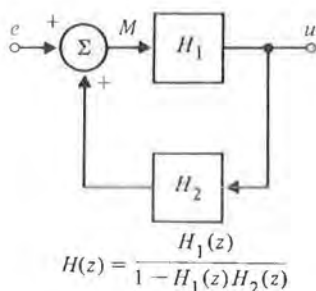


Figure 2.7 Feedback transfer function.

We can write (2.18) as

$$z^3\xi = e - a_1z^2\xi - a_2z\xi - a_3\xi$$

$$\xi(k+3) = e(k) - a_1\xi(k+2) - a_2\xi(k+1) - a_3\xi(k). \quad (2.20)$$

Now assume we have $z^3\xi$, which is to say that we have $\xi(k+3)$ because z^3 is an advance operator of three steps. If we operate on this with z^{-1} three times in a row, we will get back to $\xi(k)$, as shown in Fig. 2.8(a). From (2.20), we can now compute $z^3\xi$ from e and the lower powers of z and ξ given in the block diagram; the picture is now as given in Fig. 2.8(b). To complete the representation of (2.18) and (2.19), we need only add the formation of the output u as a weighted sum of the variables $z^3\xi$, $z^2\xi$, $z\xi$, and ξ according to (2.19). The completed picture is shown in Fig. 2.8(c).

In Fig 2.8(c), the internal variables have been named x_1 , x_2 , and x_3 . These variables comprise the *state* of this dynamic system in this form. Having the block diagram shown in Fig. 2.8(c), we can write down, almost by inspection, the difference equations that describe the evolution of the state, again using the fact that the transfer function z^{-1} corresponds to a one-unit delay. For example, we see that $x_3(k+1) = x_2(k)$ and $x_2(k+1) = x_1(k)$. Finally, expressing the sum at the far left of the figure, we have

$$x_1(k+1) = -a_1x_1(k) - a_2x_2(k) - a_3x_3(k) + e(k).$$

We collect these three equations together in proper order, and we have

$$\begin{aligned} x_1(k+1) &= -a_1x_1(k) - a_2x_2(k) - a_3x_3(k) + e(k), \\ x_2(k+1) &= x_1(k), \\ x_3(k+1) &= x_2(k). \end{aligned} \quad (2.21)$$

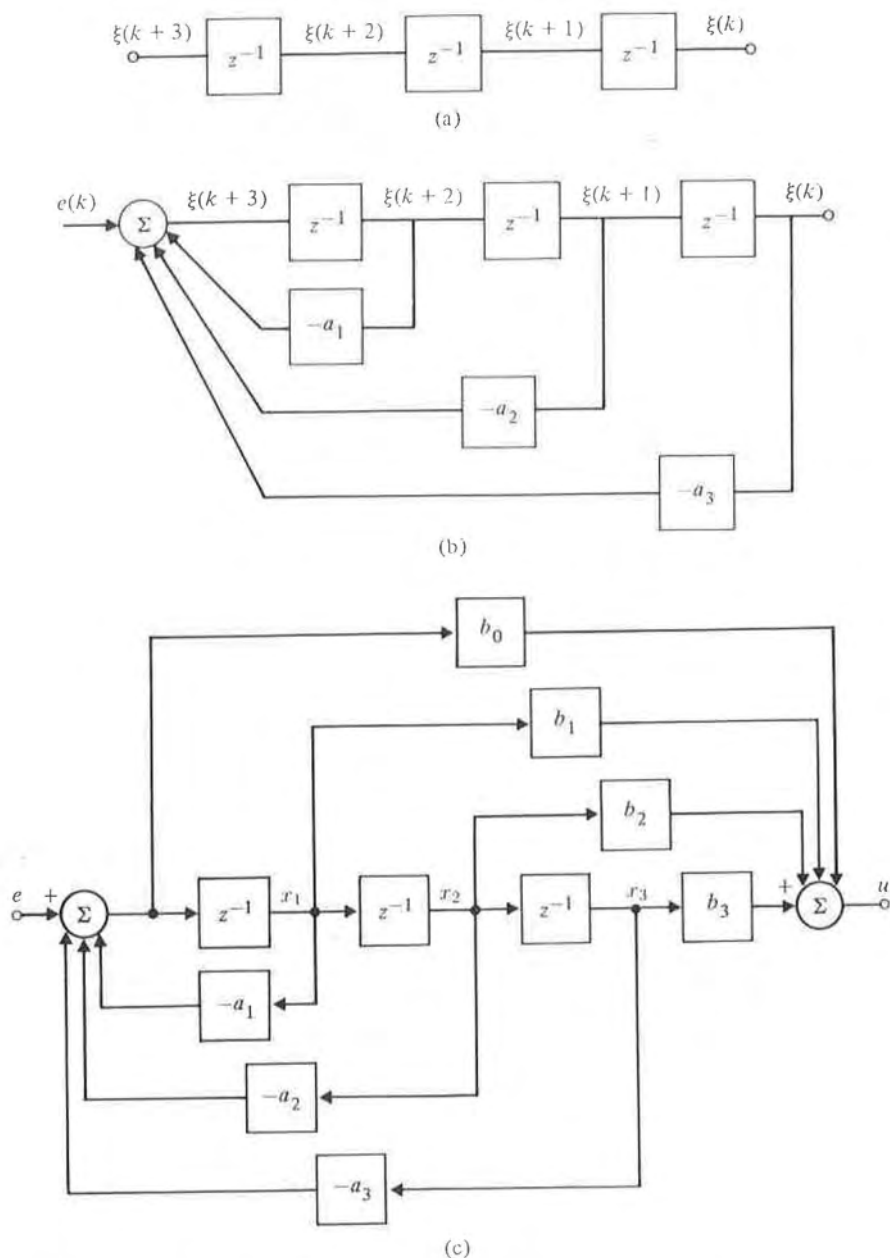


Figure 2.8 Block diagram development of control canonical form. (a) Solving for $\xi(k)$; (b) solving for $\xi(k+3)$ from $e(k)$ and past ξ 's; (c) solving for $U(k)$ from ξ 's.

Using vector-matrix notation,⁷ we can write this in the compact form

$$\mathbf{x}(k+1) = \mathbf{A}_c \mathbf{x}(k) + \mathbf{B}_c e(k),$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

$$\mathbf{A}_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad (2.22a)$$

and

$$\mathbf{B}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.22b)$$

The output equation is also immediate except that we must watch to catch *all* paths by which the state variables combine in the output. The problem is caused by the b_0 term. If $b_0 = 0$, then $u = b_1 x_1 + b_2 x_2 + b_3 x_3$, and the corresponding matrix form is immediate. However, if b_0 is not 0, x_1 for example not only reaches the output through b_1 but also by the parallel path with gain $-b_0 a_1$. The complete equation is

$$u = (b_1 - a_1 b_0)x_1 + (b_2 - a_2 b_0)x_2 + (b_3 - a_3 b_0)x_3 + b_0 e.$$

In vector/matrix notation, we have

$$u = \mathbf{C}_c \mathbf{x} + \mathbf{D}_c e,$$

where

$$\mathbf{C}_c = [b_1 - a_1 b_0 \quad b_2 - a_2 b_0 \quad b_3 - a_3 b_0], \quad (2.23a)$$

$$\mathbf{D}_c = [b_0]. \quad (2.23b)$$

⁷We assume the reader has some knowledge of matrices. The results we require and references to study material are given in Appendix C. To distinguish vectors and matrices from scalar variables, we will use bold-face type.

We can combine the equations for the state evolution and the output to give the very useful and most compact equations for the dynamic system,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_c \mathbf{x}(k) + \mathbf{B}_c e(k), \\ u(k) &= \mathbf{C}_c \mathbf{x}(k) + \mathbf{D}_c e(k), \end{aligned} \quad (2.24)$$

where \mathbf{A}_c and \mathbf{B}_c for this control canonical form are given by (2.22), and \mathbf{C}_c and \mathbf{D}_c are given by (2.23).

The other canonical form we want to illustrate is called the observer canonical form and is found by starting with the difference equations in operator/transform form as

$$z^3 u + a_1 z^2 u + a_2 z u + a_3 u = b_0 z^3 e + b_1 z^2 e + b_2 z e + b_3 e.$$

In this equation, the external input is $e(k)$, and the response is $u(k)$, which is the solution of this equation. The terms with factors of z are time-shifted toward the future with respect to k and must be eliminated in some way. To do this, we assume at the start that we have the $u(k)$, and of course the $e(k)$, and we rewrite the equation as

$$b_3 e - a_3 u = z^3 u + a_1 z^2 u + a_2 z u - b_0 z^3 e - b_1 z^2 e - b_2 z e.$$

Here, *every* term on the right is multiplied by at least one power of z , and thus we can operate on the lot by z^{-1} as shown in the partial block diagram drawn in Fig. 2.9(a).

Now in this internal result there appear $a_2 u$ and $-b_2 e$, which can be cancelled by adding proper multiples of u and e , as shown in Fig. 2.9(b), and once they have been removed, the remainder can again be operated on by z^{-1} .

If we continue this process of subtracting out the terms at k and operating on the rest by z^{-1} , we finally arrive at the place where all that is left is u alone! But that is just what we assumed we had in the first place, so connecting this term back to the start finishes the block diagram, which is drawn in Fig. 2.9(c).

A preferred choice of numbering for the state components is also shown in the figure. Following the technique used for the control form, we find that the matrix equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_0 \mathbf{x}(k) + \mathbf{B}_0 e(k), \\ u(k) &= \mathbf{C}_0 \mathbf{x}(k) + \mathbf{D}_0 e(k), \end{aligned} \quad (2.25)$$

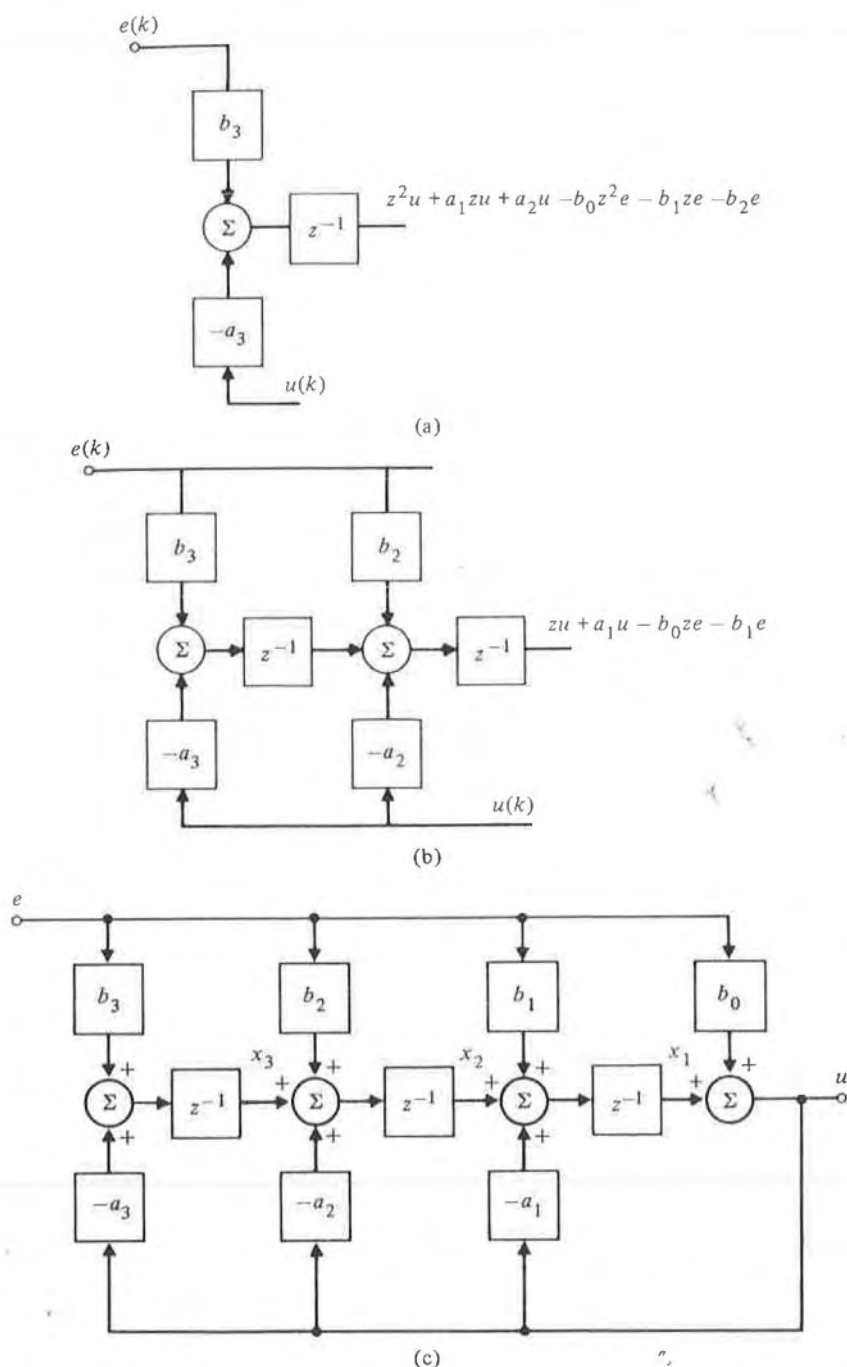


Figure 2.9 Block diagram development of observer canonical form. (a) The first partial sum and delay; (b) the second partial sum and delay; (c) the completion with the solution for $u(k)$.

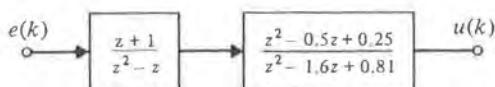


Figure 2.10 Block diagram of a cascade realization.

where

$$\mathbf{A}_0 = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_0 = \begin{bmatrix} b_1 - b_0 a_1 \\ b_2 - b_0 a_2 \\ b_3 - b_0 a_3 \end{bmatrix},$$

$$\mathbf{C}_0 = [1 \quad 0 \quad 0],$$

$$\mathbf{D}_0 = [b_0].$$

The block diagrams of Figs. 2.8 and 2.9 are called *direct canonical* realizations of the transfer function $H(z)$ because the gains of the realizations are coefficients in the transfer-function polynomials. Another useful form is obtained if we realize a transfer function by placing several first- or second-order direct forms in series with each other, a *cascade canonical* form. In this case, the $H(z)$ is represented as a product of factors, and the poles and zeros of the transfer function are clearly represented in the coefficients.

For example, suppose we have a transfer function

$$\begin{aligned} H(z) &= \frac{z^3 + 0.5z^2 - 0.25z + 0.25}{z^4 - 2.6z^3 + 2.4z^2 - 0.8z} \\ &= \frac{(z+1)(z^2 - 0.5z + 0.25)}{(z^2 - z)(z^2 - 1.6z + 0.8)}. \end{aligned}$$

The zero factor $z+1$ can be associated with the pole factor $z^2 - z$ to form one second-order system, and the zero factor $z^2 - 0.5z + 0.25$ can be associated with the second-order pole factor $z^2 - 1.6z + 0.8$ to form another. The cascade factors, which could be realized in a direct form such as control or observer form, make a cascade form as shown in Fig. 2.10.

2.3.4 Relation of Transfer Function to Pulse Response

We have shown that a transfer function of z^{-1} is a unit delay in the time domain. We can also give a time-domain meaning to an arbitrary transfer func-

Table 2.1 Step-by-step construction of the unit pulse response for Fig. 2.4.

k	e_{k-1}	e_k	a_k	u_{k-1}	$u_k \equiv h_k$
0	0	1	$T/2$	0	$T/2$
1	1	0	$T/2$	$T/2$	T
2	0	0	0	T	T
3	0	0	0	T	T

tion. Recall that the z -transform is defined by (2.8) to be $E(z) = \sum e_k z^{-k}$, and the transfer function is defined from (2.16) as $H(z)$ when the input and output are related by $U(z) = H(z)E(z)$. Now suppose we deliberately select $e(k)$ to be the unit discrete pulse defined by

$$e_k = \begin{cases} 1, & (k = 0), \\ 0, & (k \neq 0), \end{cases} \triangleq \delta_k. \quad (2.26)$$

Then it follows that $E(z) = 1$ and therefore that

$$U(z) = H(z). \quad (2.27)$$

Thus the transfer function $H(z)$ is seen to be the *transform* of the response to a unit-pulse input. For example, let us look at the system of Fig. 2.4 and put a unit pulse in at the e_k -node (with no signals in the system beforehand).⁸ We can readily follow the pulse through the block and build Table 2.1.

Thus the unit-pulse response is zero for negative k , is $T/2$ at $k = 0$, and equals T thereafter. The z -transform of this sequence is

$$H(z) = \sum_{-\infty}^{\infty} u_k z^{-k} \triangleq \sum_{-\infty}^{\infty} h_k z^{-k}.$$

⁸In this development we assume that (2.7) is intended to be used as a formula for computing values of u_k as k increases. There is no reason why we could not also solve for u_k as k takes on negative values. The direction of time comes from the application and not from the recurrence equation.

e_0	$+e_1 z^{-1}$	$+e_2 z^{-2}$	$+e_3 z^{-3}$	$+\dots$
h_0	$+h_1 z^{-1}$	$+h_2 z^{-2}$	$+h_3 z^{-3}$	$+\dots$
$e_0 h_0$	$+e_1 h_0 z^{-1}$	$+e_2 h_0 z^{-2}$	$+e_3 h_0 z^{-3}$	
	$+e_0 h_1 z^{-1}$	$+e_1 h_1 z^{-2}$	$+e_2 h_1 z^{-3}$	
		$+e_0 h_2 z^{-2}$	$+e_1 h_2 z^{-3}$	
			$+e_0 h_3 z^{-3}$	
$e_0 h_0 + (e_0 h_1 + e_1 h_0) z^{-1} + (e_0 h_2 + e_1 h_1 + e_2 h_0) z^{-2} + (e_0 h_3 + e_1 h_2 + e_2 h_1 + e_3 h_0) z^{-3} + \dots$				

Figure 2.11 Representation of the product $E(z)H(z)$ as a product of polynomials.

If we add $T/2$ to the z^0 -term and subtract $T/2$ from the whole series, we have a simpler sum, as follows:

$$\begin{aligned}
 H(z) &= \sum_{k=0}^{\infty} T z^{-k} - \frac{T}{2} \\
 &= \frac{T}{1 - z^{-1}} - \frac{T}{2} \quad (1 < |z|) \\
 &= \frac{2T - T(1 - z^{-1})}{2(1 - z^{-1})} \\
 &= \frac{T + T z^{-1}}{2(1 - z^{-1})} \\
 &= \frac{T z + 1}{2 z - 1} \quad (1 < |z|). \quad (2.28)
 \end{aligned}$$

Of course, this is the transfer function we obtained in (2.13) from direct analysis of the difference equation.

A final point of view useful in the interpretation of the discrete transfer function is obtained by multiplying the infinite polynomials of $E(z)$ and $H(z)$ as suggested in (2.16). For purposes of illustration, we will assume that the unit-pulse response, h_k , is zero for $k < 0$. Likewise, we will take $k = 0$ to be the starting time for e_k . Then the product that produces $U(z)$ is the polynomial product given in Fig. 2.11.

Since this product has been shown to be $U(z) = \sum u_k z^{-k}$, it must therefore follow that the coefficient of z^{-k} in the product is u_k . Listing these

coefficients, we have the relations

$$\begin{aligned}u_0 &= e_0 h_0, \\u_1 &= e_0 h_1 + e_1 h_0, \\u_2 &= e_0 h_2 + e_1 h_1 + e_2 h_0, \\u_3 &= e_0 h_3 + e_1 h_2 + e_2 h_1 + e_3 h_0.\end{aligned}$$

The extrapolation of this simple pattern gives the result

$$u_k = \sum_{j=0}^k e_j h_{k-j}.$$

By extension, we let the lower limit of the sum be $-\infty$ and the upper limit be $+\infty$:

$$u_k = \sum_{j=-\infty}^{\infty} e_j h_{k-j}. \quad (2.29)$$

Negative values of j in the sum correspond to inputs applied before time equals zero. Values for j greater than k occur if the unit-pulse response is nonzero for negative arguments. By definition, such a system, which responds *before* the input that causes it occurs, is called *noncausal*. This is the discrete convolution sum and is the analog of the convolution integral that relates input and impulse response to output in linear, constant, continuous systems.

To verify (2.29) we can take the z -transform of both sides:

$$\sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=-\infty}^{\infty} z^{-k} \sum_{j=-\infty}^{\infty} e_j h_{k-j}.$$

Interchanging the sum on j with the sum on k leads to

$$U(z) = \sum_{j=-\infty}^{\infty} e_j \sum_{k=-\infty}^{\infty} z^{-k} h_{k-j}.$$

Now let $k - j = l$ in the second sum:

$$U(z) = \sum_{j=-\infty}^{\infty} e_j \sum_{l=-\infty}^{\infty} h_l z^{-(l+j)},$$

but $z^{-(l+j)} = z^{-l}z^{-j}$, which leads to

$$U(z) = \sum_{j=-\infty}^{\infty} e_j z^{-j} \sum_{l=-\infty}^{\infty} h_l z^{-l},$$

and we recognize these two separate sums as

$$U(z) = E(z)H(z).$$

We can also derive the convolution sum from the properties of linearity and stationarity. First we need more formal definitions of "linear" and "stationary."

1. *Linearity*: A system with input e and output u is *linear* if superposition applies, which is to say, if $u_1(k)$ is the response to $e_1(k)$ and $u_2(k)$ is the response to $e_2(k)$, then the system is linear if and only if, for every scalar α and β , the response to $\alpha e_1 + \beta e_2$ is $\alpha u_1 + \beta u_2$.
2. *Stationarity*: A system is *stationary*, or time invariant, if a time shift in the input results in only a time shift in the output. For example, if we take the system at rest (no internal energy in the system) and apply a certain signal $e(k)$, suppose we observe a response $u(k)$. If we repeat this experiment at any later time when the system is again at rest and we apply the shifted input, $e(k - N)$, if we see $u(k - N)$, then the system is stationary.

These properties can be used to derive the convolution in (2.29) as follows. If response to a unit pulse at $k = 0$ is $h(k)$, then response to a pulse of intensity e_0 is $e_0 h(k)$ if the system is linear. Furthermore, if the system is *constant*, then a delay of the input will delay the response. Thus, if

$$e = \begin{cases} e_l, & k = l, \\ 0, & k \neq l, \end{cases}$$

then the response will be $e_l h_{k-l}$.

Finally, by linearity again, the total response at time k to a sequence of these pulses is the *sum* of the responses, namely,

$$u_k = e_0 h_k + e_1 h_{k-1} + \cdots + e_l h_{k-l} + \cdots + e_k h_0,$$

or

$$u_k = \sum_{l=0}^k e_l h_{k-l}.$$

Now note that if the input sequence began in the distant past, we must include terms for $l < 0$, perhaps back to $l = -\infty$. Similarly, if the system should be noncausal, future values of e where $l > k$ may also come in. The general case is thus (again)

$$u_k = \sum_{l=-\infty}^{\infty} e_l h_{k-l}. \quad (2.30)$$

2.3.5 External Stability and Jury's Test

A very important qualitative property of a dynamic system is stability, and we can consider internal or external stability. Internal stability is concerned with the responses at all the internal variables such as those that appear at the delay elements in a canonical block diagram as in Fig. 2.8 or Fig. 2.9 (the state). Otherwise we can be satisfied to consider only the *external stability* as given by the study of the input-output relation described for the linear stationary case by the convolution (2.30). These differ in that some internal modes might not be connected to both the input and the output of a given system.

For external stability, the most common definition of *appropriate response* is that for every Bounded Input, we should have a Bounded Output. If this is true we say the system is BIBO stable. A test for BIBO stability can be given directly in terms of the unit-pulse response, h_k . First we consider a sufficient condition. Suppose the input e_k is bounded, that is, there is an M such that

$$|e_l| \leq M < \infty \quad \text{for all } l. \quad (2.31)$$

If we consider the magnitude of the response given by (2.30), it is easy to see that

$$|u_k| \leq \left| \sum e_l h_{k-l} \right|,$$

which is surely less than the sum of the magnitudes as given by

$$\leq \sum_{-\infty}^{\infty} |e_l| |h_{k-l}|.$$

But, because we assume (2.31), this result is in turn bounded by

$$\leq M \sum_{-\infty}^{\infty} |h_{k-l}|. \quad (2.32)$$

Thus the output will be bounded for every bounded input if

$$\sum_{l=-\infty}^{\infty} |h_{k-l}| < \infty. \quad (2.33)$$

This condition is also necessary, for if we consider the bounded (by 1!) input

$$\begin{aligned} e_l &= \frac{h_{-l}}{|h_{-l}|} & (h_{-l} \neq 0) \\ &= 0 & (h_{-l} = 0) \end{aligned}$$

and apply it to (2.30), the output at $k = 0$ is

$$\begin{aligned} u_0 &= \sum_{l=-\infty}^{\infty} e_l h_{-l} \\ &= \sum_{l=-\infty}^{\infty} \frac{(h_{-l})^2}{|h_{-l}|} \\ &= \sum_{l=-\infty}^{\infty} |h_{-l}|. \end{aligned} \quad (2.34)$$

Thus, unless the condition given by (2.34) is true, the system is not BIBO stable.

Example 2.3: The test given by (2.34) can be applied to the unit pulse response used to compute (2.13) and given as the u_k -column in Table 2.1 on page 31:

$$\begin{aligned} h_0 &= T/2, \\ h_k &= T, \quad k > 0, \\ \sum |h_k| &= T/2 + \sum_{k=1}^{\infty} T = \text{unbounded}. \end{aligned} \quad (2.35)$$

Thus this discrete approximation to integration is not (BIBO) stable!

Example 2.4: As a second example, we consider the difference equation (2.2) with all coefficients except a_1 and b_0 equal to zero:

$$u_k = a_1 u_{k-1} + b_0 e_k. \quad (2.36)$$

The unit-pulse response is easily developed from the first few terms to be

$$\begin{aligned} u_0 &= b_0, & u_1 &= a_1 b_0, & u_2 &= a_1^2 b_0, \dots \\ u_k &= h_k = b_0 a^k, & k &\geq 0. \end{aligned} \quad (2.37)$$

Applying the test, we have

$$\begin{aligned} \sum_{-\infty}^{\infty} |h_l| &= \sum_{\infty=0}^{\infty} b_0 |a^l| = b_0 \frac{1}{1 - |a|} & (|a| < 1) \\ &= \text{unbounded} & (|a| \geq 1). \end{aligned}$$

Thus we conclude that the system described by this equation is BIBO stable if $|a| < 1$, and unstable otherwise.

For a more general rational transfer function with many simple poles, we can expand the function in partial fractions about its poles, and the corresponding pulse response will be a sum of respective terms. As we saw earlier, if a pole is inside the unit circle, the corresponding pulse response decays with time geometrically and is stable. Thus, if all poles are inside the unit circle, the system with rational transfer function is stable; if at least one pole is on or outside the unit circle, the corresponding system is not BIBO stable. With modern computer programs available, finding the poles of a particular transfer function is no big deal. Sometimes, however, we wish to test for stability of an entire class of systems; or, as in an adaptive control system, the potential poles are constantly changing and we wish to have a quick test for stability in terms of the literal polynomial coefficients. In the continuous case, such a test was provided by Routh; in the discrete case, the most convenient such test was worked out by Jury and Blanchard (1961).

The Jury test is in the same spirit as the Routh test (see Franklin, Powell, and Emami-Naeini for a discussion) in that we form two rows from the coefficients of length n , and from these, by a series of two-by-two determinants, we compute a successor row of length $n - 1$. With this reduced

length row we form another successor of length $n - 2$ and so on until we have a row of length 1. The test consists of examining the sign of the first entries in selected rows. As with the Routh test, the Jury test is much more difficult to derive than to use; here we illustrate only the use.

If we have a transfer function $H(z) = b(z)/a(z)$, then this system will be stable if and only if all roots of $a(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n$ are inside the unit circle. To test for this condition by the Jury test, multiply $a(z)$ by -1 if necessary to make the sign of a_0 positive. Then form rows of the coefficients, the even rows being in reversed order, as follows:

$$\begin{array}{cccc} a_0 & a_1 & \dots & a_n \\ a_n & a_{n-1} & \dots & a_0 \\ b_0 & b_1 & \dots & . \\ b_{n-1} & b_{n-2} & \dots & . \end{array}$$

The entries in the third row are formed from the second-order determinants using the first column of the first two rows with each of the other columns from these rows *starting from the right* and dividing by a_0 . The result can be expressed by the formulas:

$$\begin{aligned} b_0 &= a_0 - \frac{a_n}{a_0} a_n, \\ b_1 &= a_1 - \frac{a_n}{a_0} a_{n-1}, \\ b_k &= a_k - \frac{a_n}{a_0} a_{n-k}, \end{aligned}$$

The elements in the third row are reversed to form the fourth row and the process is repeated. For example, the elements of the fifth row are given by

$$c_k = b_k - \frac{b_{n-1}}{b_0} b_{n-1-k}.$$

The original polynomial is stable (has all roots inside the unit circle) if all the terms in the first columns of the odd rows are positive, that is, if $a_0 > 0, b_0 > 0, c_0 > 0, \dots$

This test is readily implemented in a computer program.⁹

⁹See STABLE in Table E.1 in Appendix E.

Example 2.5: To illustrate the use of Jury's test, we consider first the simple second-order polynomial

$$a(z) = z^2 + a_1z + a_2.$$

The Jury array is

$$\begin{array}{ccc} 1 & a_1 & a_2 \\ a_2 & a_1 & 1 \\ 1 - a_2^2 & a_1 - a_1a_2 & \cdot \\ a_1 - a_1a_2 & 1 - a_2^2 & \cdot \\ \frac{(1 - a_2^2)^2 - a_1^2(1 - a_2)^2}{1 - a_2^2} & \cdot & \cdot \end{array}$$

From row three, we have the condition that $1 - a_2^2 > 0$, and from this we conclude that

$$-1 < a_2 < 1.$$

From row five, we can factor out $(1 - a_2)^2$ to conclude that

$$(1 + a_2)^2 > a_1^2$$

and thus,

$$a_2 + 1 > a_1 \quad \text{and} \quad a_2 + 1 > -a_1.$$

From these inequalities, we can draw the *stability triangle* shown in Fig 2.12.

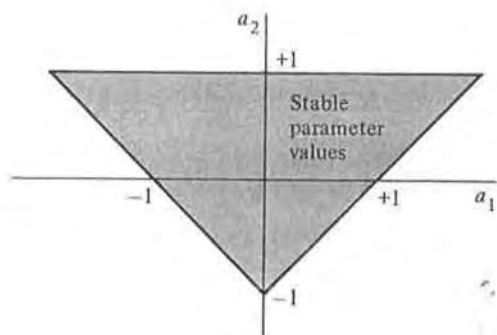


Figure 2.12 Stability triangle for the general, real, second-order polynomial.

Example 2.6: For the next example, consider the polynomial

$$a(z) = z^3 - 2.1z^2 + 1.6z - 0.4.$$

The Jury array is

1	-2.1	1.6	-0.4
-0.4	1.6	-2.1	1
0.84	-1.46	0.76	.
0.76	-1.46	0.84	.
0.1524	-0.139	.	.
-0.139	0.1524	.	.
0.0256	.	.	.

The test is from the odd rows: $1 > 0$, $0.84 > 0$, $0.1524 > 0$, $0.0256 > 0$, and we conclude that a system with this polynomial as its denominator would be stable. As a matter of fact, the poles are at $z = 0.5$ and $z = 0.8 \pm 0.4j$.

Example 2.7: As a third example, consider the polynomial

$$z^3 - 2.6z^2 + 2.4z - 0.8$$

The odd rows only of the Jury array are

1	-2.6	2.4	-0.8
0.36	-0.68	0.32	.
0.0756	-0.0756	.	.
0	.	.	.

From these computations we conclude that the polynomial does *not* have all its roots inside the unit circle; and, because the last term is zero and a small perturbation would send it either way, inside or outside, there must be at least one root exactly on the unit circle. In fact, the roots are $z = 1$ and $z = 0.8 \pm 0.4j$.

As an aid to testing stability, it can be shown that it is *necessary* for a stable polynomial (with positive first term) that the polynomial evaluated

at $z = 1$ and $z = -1$ must both be positive. The first value is just the sum of the coefficients, and the second is the sum with alternating sign changes. These two tests are quickly done and can save time if the only purpose is to be sure that the system is stable; many unstable systems will be rejected by these simple tests without going through the entire Jury array. In the previous case, for instance, the sum of coefficients is zero and we need go no further; the polynomial cannot be stable.

2.4 DISCRETE MODELS OF SAMPLED-DATA SYSTEMS

The systems and signals we have studied thus far have been defined in discrete time only. Most of the dynamic systems to be controlled, however, are continuous systems and, if linear, are described by continuous transfer functions in the Laplace variable s . The interface between the continuous and discrete domains are the A/D and the D/A converters as shown in Fig. 1.1. In this section we develop the analysis needed to compute the discrete transfer function between the samples that come from the digital computer to the D/A converter and the samples that are picked up by the A/D converter.¹⁰ The situation is drawn in Fig. 2.13.

2.4.1 Using the z -Transform

We wish to find the discrete transfer function from the input samples $u(kT)$ (which probably come from a computer of some kind) to the output samples, $y(kT)$ picked up by the A/D converter. Although it is possibly confusing at first, we follow convention and call the discrete transfer function $G(z)$ when the continuous transfer function is $G(s)$. Although $G(z)$ and $G(s)$ are entirely different functions, they do describe the *same* plant, and the use of s for the continuous transform and z for the discrete transform is always maintained. To find $G(z)$ we need only observe that the $y(kT)$ are samples of the plant output when the input is from the D/A converter. As for the D/A converter, we assume that this device, commonly called a zero-order hold or ZOH, accepts a sample $u(kT)$ at $t = kT$ and holds its output constant

¹⁰In Chapter 3, a comprehensive frequency analysis of sampled data systems is presented. Here we undertake only the special problem of finding the sample-to-sample discrete transfer function of a continuous system between a D/A and an A/D.

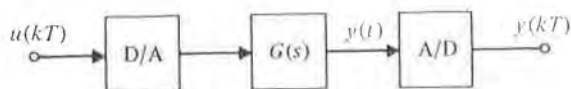


Figure 2.13 The prototype sampled-data system.

at this value until the next sample is sent at $t = kT + T$. The piecewise constant output of the D/A is the signal, $u(t)$, that is applied to the plant

Our problem is now really quite simple because we have just seen that the discrete transfer function is the z -transform of the samples of the output when the input samples are the unit pulse at $k = 0$. If $u(kT) = 1$ for $k = 0$ and $u(kT) = 0$ for $k \neq 0$, the output of the D/A converter is a pulse of width T seconds and height 1, as sketched in Fig. 2.14. Mathematically, this pulse is given by $1(t) - 1(t - T)$. Let us call the particular output in response to the pulse shown in Fig. 2.14 $y_1(t)$. This response is the difference between the step response [to $1(t)$] and the delayed step response [to $1(t - T)$]. The Laplace transform of the step response is $G(s)/s$. Thus in the transform domain the unit pulse response of the plant is

$$Y_1(s) = (1 - e^{-Ts}) \frac{G(s)}{s}, \quad (2.38)$$

and the required transfer function is the z -transform of the samples of the inverse of $Y_1(s)$, which can be expressed as

$$\begin{aligned} G(z) &= \mathcal{Z}\{y_1(kT)\} \\ &= \mathcal{Z}\{\mathcal{L}^{-1}\{Y_1(s)\}\} \triangleq \mathcal{Z}\{Y_1(s)\} \\ &= \mathcal{Z}\left\{(1 - e^{-Ts}) \frac{G(s)}{s}\right\}. \end{aligned}$$

This is the sum of two parts. The first is $\mathcal{Z}\{G(s)/s\}$, and the second is

$$\mathcal{Z}\{e^{-Ts}G(s)/s\} = z^{-1}\mathcal{Z}\{G(s)/s\}$$

because e^{-Ts} is exactly a delay of one period. Thus the transfer function

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\} \quad (2.39)$$

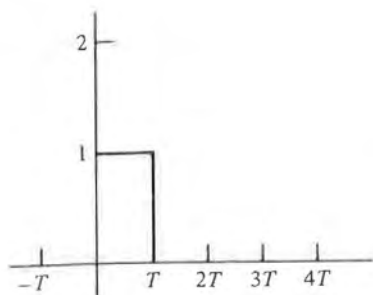


Figure 2.14 D/A output for unit-pulse input.

Example 2.8: As a first example of computing such discrete transfer functions, suppose $G(s) = a/(s + a)$. Then

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

and the corresponding time function is

$$\mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} = 1(t) - e^{-at}1(t).$$

The samples of this signal are $1(kT) - e^{-akT}1(kT)$, and the z -transform of these samples is

$$\begin{aligned} \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} &= \frac{z}{z-1} - \frac{z}{z-e^{-aT}} \\ &= \frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}. \end{aligned}$$

We could have gone to the tables in Appendix B and found this result directly as entry 12. Now we can compute the desired transform by applying (2.39)

$$\begin{aligned} G(z) &= \frac{z-1}{z} \frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})} \\ &= \frac{1-e^{-aT}}{z-e^{-aT}}. \end{aligned} \tag{2.40}$$

Example 2.9: We consider the double integrator characteristic of a single mass, such as the satellite, for which the transfer function is $G(s) = 1/s^2$. We have

$$G(z) = (1 - z^{-1})Z\left\{\frac{1}{s^3}\right\}$$

This time we refer to the tables in Appendix B and find that the transform of $1/s^3$ is

$$\frac{T^2}{2} \frac{z(z+1)}{(z-1)^3},$$

and therefore

$$G(z) = \frac{T^2(z+1)}{2(z-1)^2}. \quad (2.41)$$

For more complex systems than those in Examples 2.8 and 2.9, use of a CAD package is recommended.¹¹

2.4.2 Continuous Time Delay

We now consider computing the discrete transfer function of a continuous system with pure time delay. The responses of many chemical process-control plants exhibit pure time delay because there is a finite time of transport of fluids or materials between the process and the controls and/or the sensors. Also, we must often consider finite computation time in the digital controller, and this is exactly the same as if the process had a pure time delay. With the techniques we have developed here, it is possible to obtain the discrete transfer function of such processes exactly, as Example 2.10 illustrates.

Example 2.10: We consider the example suggested by the fluid mixer problem described in Appendix A.3, for which

$$G(s) = e^{-\lambda s} H(s).$$

¹¹See X-C2D in Table E.1.

The term $e^{-\lambda s}$ represents the delay of λ seconds, which includes both the process delay and the computation delay, if any. We assume that $H(s)$ is a rational transfer function. To prepare this function for computation of the z -transform, we first define an integer ℓ and a positive number m less than 1.0 such that $\lambda = \ell T - mT$. With these definitions we can write

$$\frac{G(s)}{s} = e^{-\ell T s} \frac{e^{mT s} H(s)}{s}.$$

Because ℓ is an integer, this term reduces to $z^{-\ell}$ when we take the z -transform. Because $m < 1$, the transform of the other term is quite direct. We select $H(s) = a/(s + a)$ and, after the partial fraction expansion of $H(s)/s$, we have

$$G(z) = \frac{z-1}{z^{\ell+1}} \mathcal{Z} \left\{ \frac{e^{mT s}}{s} - \frac{e^{mT s}}{s+a} \right\}$$

To complete the transfer function, we need the z -transforms of the inverses of the terms in the braces. The first term is a unit step shifted left by mT seconds, and the second term is an exponential shifted left by the same amount. Because $m < 1$, these shifts are less than one full period, and no sample is picked up in negative time. The signals are sketched in Fig. 2.15.

The samples are given by $1(kT)$ and $e^{-aT(k+m)}1(kT)$. The corresponding z -transforms are $z/(z-1)$ and $ze^{-amT}/(z-e^{-aT})$. Consequently the final transfer function is

$$\begin{aligned} G(z) &= \frac{z-1}{z} \frac{1}{z^{\ell}} \left\{ \frac{z}{z-1} - \frac{ze^{-amT}}{z-e^{-aT}} \right\} \\ &= \frac{z-1}{z^{\ell}} \left\{ \frac{z[z-e^{-aT} - (z-1)e^{-amT}]}{(z-1)(z-e^{-aT})} \right\} \\ &= (1-e^{-amT}) \frac{z+\alpha}{z^{\ell}(z-e^{-aT})}, \end{aligned}$$

where the zero position is at $-\alpha = -(e^{-amT} - e^{-aT})/(1 - e^{-amT})$. Notice that this zero is near the origin of the z -plane when m is near 1 and moves outside the unit circle to near $-\infty$ when m approaches 0. For the specific values of the mixer, we take $a = 1$, $T = 1$, and

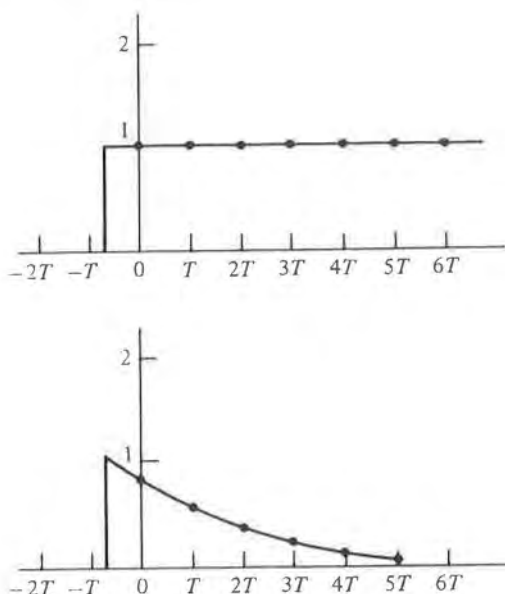


Figure 2.15 Sketch of the shifted signals showing sample points.

$\lambda = 1.5$. Then we can compute that $\ell = 2$ and $m = 0.5$. For these values, we get

$$G(z) = \frac{z + 0.6065}{z^2(z - 0.3679)}. \quad (2.42)$$

2.4.3 State-Space Form

Computing the z -transform using the Laplace transform as in (2.39) is a very tedious business that is unnecessary with the availability of computers. We will next develop a formula using state descriptions that will remove most of the calculations to the computer, where it is better done. A continuous, linear, constant-coefficient system of differential equations can always be expressed as a set of first-order matrix differential equations:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u + \mathbf{G}_1w, \quad (2.43)$$

where u is the control input to the system and w is a disturbance input. The output can be expressed as a linear combination of the state, \mathbf{x} , and the input as

$$y = \mathbf{H}\mathbf{x} + Ju. \quad (2.44)$$

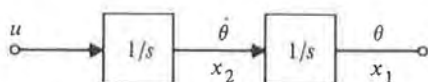


Figure 2.16 Satellite attitude control in classical representation.

Often the sampled-data system being described is the plant of a control problem, and the parameter J in (2.44) is zero and will frequently be omitted.

Example 2.11: Application of state representation to the equations of the satellite attitude-control example shown in Fig. 2.16 and described in Appendix A yields

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{F}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{G}} u, \\ \theta = y &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned} \quad (2.45)$$

which, in this case, turns out to be a rather involved way of writing

$$\ddot{\theta} = u.$$

The representations (2.43) and (2.44) are not unique. Given one state representation, any nonsingular linear transformation of that state such as $\xi = \mathbf{T}\mathbf{x}$ is also an allowable alternative realization of the same system.

If we let $\xi = \mathbf{T}\mathbf{x}$ in (2.43) and (2.44), we find

$$\begin{aligned} \dot{\xi} &= \mathbf{T}\dot{\mathbf{x}} = \mathbf{T}(\mathbf{F}\mathbf{x} + \mathbf{G}u + \mathbf{G}_1w) \\ &= \mathbf{T}\mathbf{F}\mathbf{x} + \mathbf{T}\mathbf{G}u + \mathbf{T}\mathbf{G}_1w, \\ \dot{\xi} &= \mathbf{T}\mathbf{F}\mathbf{T}^{-1}\xi + \mathbf{T}\mathbf{G}u + \mathbf{T}\mathbf{G}_1w, \\ y &= \mathbf{H}\mathbf{T}^{-1}\xi + Ju. \end{aligned}$$

If we designate the system matrices for the new state ξ as \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} , then

$$\dot{\xi} = \mathbf{A}\xi + \mathbf{B}u + \mathbf{B}_1w, \quad y = \mathbf{C}\xi + \mathbf{D}u,$$

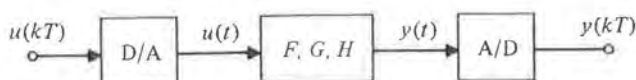


Figure 2.17 System definition with sampling operations shown.

where

$$A = TFT^{-1}, \quad B = TG, \quad B_1 = TG_1, \quad C = HT^{-1}, \quad D = J.$$

Example 2.12: As an illustration, we can let $\xi_1 = x_2$ and $\xi_2 = x_1$ in (2.45); or, in matrix notation, the transformation to interchange the states is

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

In this case $T^{-1} = T$, and application of the transformation equations to the system matrices of (2.45) gives

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \quad 1].$$

Most often, a change of state is made to bring the description matrices into a useful canonical form. We saw earlier how a single high-order difference equation could be represented by a state description in control or in observer canonical form. Also, there is a very useful state description corresponding to the partial-fraction expansion of a transfer function. State transformations can take a general description for either a continuous or a discrete system and, subject to some technical restrictions, convert it into a description in one or the other of these forms, as needed.

We wish to use the state description to establish a general method for obtaining the difference equations that represent the behavior of the continuous plant. Fig. 2.17 again depicts the portion of our system under consideration. Ultimately, the digital controller will take the samples $y(k)$, operate on that sequence by means of a difference equation, and put out a sequence of numbers, $u(k)$, which are the inputs to the plant. The loop will, therefore, be closed. To analyze the result, we must be able to relate the samples of the output $y(k)$ to the samples of the control $u(k)$. To do this, we must solve (2.43).

We will solve the general equation in two steps. We begin by solving the equation with only initial conditions and no external input. This is the homogeneous equation

$$\dot{\mathbf{x}}_h = \mathbf{F}\mathbf{x}_h(t), \quad \mathbf{x}_h(t_0) = \mathbf{x}_0 \quad (2.46)$$

To solve this, we assume the solution is sufficiently smooth that a series expansion of the solution is possible:

$$\mathbf{x}_h(t) = \mathbf{A}_0 + \mathbf{A}_1(t - t_0) + \mathbf{A}_2(t - t_0)^2 + \dots \quad (2.47)$$

If we let $t = t_0$, we find immediately that $\mathbf{A}_0 = \mathbf{x}_0$. If we differentiate (2.47) and substitute into (2.46), we have

$$\mathbf{A}_1 + 2\mathbf{A}_2(t - t_0) + 3\mathbf{A}_3(t - t_0)^2 + \dots = \mathbf{F}\mathbf{x}_h$$

and, at $t = t_0$, $\mathbf{A}_1 = \mathbf{F}\mathbf{x}_0$. Now we continue to differentiate the series and the differential equation and equate them at t_0 to arrive at the series

$$\mathbf{x}_h(t) = \left[\mathbf{I} + \mathbf{F}(t - t_0) + \frac{\mathbf{F}^2(t - t_0)^2}{2} + \frac{\mathbf{F}^3(t - t_0)^3}{6} + \dots \right] \mathbf{x}_0.$$

This series is defined as the matrix exponential and written

$$\mathbf{x}_h(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}(t_0), \quad (2.48)$$

where, by definition, the matrix exponential is

$$\begin{aligned} e^{\mathbf{F}(t-t_0)} &= \mathbf{I} + \mathbf{F}(t - t_0) + \mathbf{F}^2 \frac{(t - t_0)^2}{2!} + \mathbf{F}^3 \frac{(t - t_0)^3}{3!} + \dots \\ &= \sum_{k=0}^{\infty} \mathbf{F}^k \frac{(t - t_0)^k}{k!}. \end{aligned} \quad (2.49)$$

It can be shown that the solution given by (2.48) is unique, which leads to very interesting properties of the matrix exponential. For example, consider two values of t : t_1 and t_2 . We have

$$\mathbf{x}(t_1) = e^{\mathbf{F}(t_1-t_0)} \mathbf{x}(t_0)$$

and

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_0)} \mathbf{x}(t_0).$$

Because t_0 is arbitrary also, we can express $\mathbf{x}(t_2)$ as if the equation solution began at t_1 , for which

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_1)}\mathbf{x}(t_1).$$

Substituting for $\mathbf{x}(t_1)$ gives

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_1)}e^{\mathbf{F}(t_1-t_0)}\mathbf{x}(t_0).$$

We now have two separate expressions for $\mathbf{x}(t_2)$, and, if the solution is unique, these must be the same. Hence we conclude that

$$e^{\mathbf{F}(t_2-t_0)} = e^{\mathbf{F}(t_2-t_1)}e^{\mathbf{F}(t_1-t_0)} \quad (2.50)$$

for all t_2, t_1, t_0 . Note especially that if $t_2 = t_0$, then

$$\mathbf{I} = e^{-\mathbf{F}(t_1-t_0)}e^{\mathbf{F}(t_1-t_0)}.$$

Thus we can obtain the inverse of $e^{\mathbf{F}t}$ by merely changing the sign of t ! We will use this result in computing the particular solution to (2.43).

The particular solution when u is not zero is obtained by using the method of *variation of parameters*.¹² We guess the solution to be in the form

$$\mathbf{x}_p(t) = e^{\mathbf{F}(t-t_0)}\mathbf{v}(t), \quad (2.51)$$

where $\mathbf{v}(t)$ is a vector of variable parameters to be determined [as contrasted to the constant parameters $\mathbf{x}(t_0)$ in (2.48)]. Substituting (2.51) into (2.43), we obtain

$$\mathbf{F}e^{\mathbf{F}(t-t_0)}\mathbf{v} + e^{\mathbf{F}(t-t_0)}\dot{\mathbf{v}} = \mathbf{F}e^{\mathbf{F}(t-t_0)}\mathbf{v} + \mathbf{G}u,$$

and, using the fact that the inverse is found by changing the sign of the exponent, we can solve for $\dot{\mathbf{v}}$ as

$$\dot{\mathbf{v}}(t) = e^{-\mathbf{F}(t-t_0)}\mathbf{G}u(t).$$

¹²Due to Joseph Louis Lagrange, French mathematician (1736–1813). We assume $w = 0$, but because the equations are linear, the effect of w can be added later.

Assuming that the control $u(t)$ is zero for $t < t_0$, we can integrate $\dot{\mathbf{v}}$ from t_0 to t to obtain

$$\mathbf{v}(t) = \int_{t_0}^t e^{-\mathbf{F}(\tau-t_0)} \mathbf{G}u(\tau) d\tau.$$

Hence, from (2.51), we get

$$\mathbf{x}_p(t) = e^{\mathbf{F}(t-t_0)} \int_{t_0}^t e^{-\mathbf{F}(\tau-t_0)} \mathbf{G}u(\tau) d\tau,$$

and simplifying, using the results of (2.50), we obtain the particular solution (convolution)

$$\mathbf{x}_p(t) = \int_{t_0}^t e^{\mathbf{F}(t-\tau)} \mathbf{G}u(\tau) d\tau. \quad (2.52)$$

The total solution for $w = 0$ and $u \neq 0$ is the sum of (2.48) and (2.52):

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)} \mathbf{G}u(\tau) d\tau. \quad (2.53)$$

We wish to use this solution over one sample period to obtain a difference equation: hence we juggle the notation a bit (let $t = kT + T$ and t_0 equal kT) and arrive at a particular version of (2.53):

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T} \mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+T-\tau)} \mathbf{G}u(\tau) d\tau. \quad (2.54)$$

This result is not dependent on the type of hold because u is specified in terms of its continuous time history, $u(t)$, over the sample interval. A common and typically valid assumption is that of a zero-order hold (ZOH) with no delay, that is,

$$u(\tau) = u(kT), \quad kT \leq \tau < kT + T.$$

If some other hold is implemented or if there is a delay between the application of the control from the ZOH and the sample point, this fact can be accounted for in the evaluation of the integral in (2.54). The equations for a delayed ZOH will be given in the next subsection. To facilitate the solution

of (2.54) for a ZOH with no delay, we change variables in the integral from τ to η such that

$$\eta = kT + T - \tau.$$

Then we have

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T} \mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G} u(kT). \quad (2.55)$$

If we define

$$\Phi = e^{\mathbf{F}T}, \quad (2.56a)$$

$$\Gamma = \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G}, \quad (2.56b)$$

Eqs. (2.55) and (2.44) reduce to difference equations in standard form:

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi \mathbf{x}(k) + \Gamma u(k) + \Gamma_1 w(k), \\ y(k) &= \mathbf{H} \mathbf{x}(k), \end{aligned} \quad (2.57)$$

where we include the effect of an impulsive or piecewise constant disturbance, w , and assume that $J = 0$ in this case. If w is a constant, then Γ_1 is given by (2.56b) with \mathbf{G} replaced by \mathbf{G}_1 . If w is an impulse, then $\Gamma_1 = \mathbf{G}_1$.¹³ The Φ series expansion

$$\Phi = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F}^2 T^2}{2!} + \frac{\mathbf{F}^3 T^3}{3!} + \cdots,$$

can also be written

$$\Phi = \mathbf{I} + \mathbf{F}T\Psi, \quad (2.58)$$

where

$$\Psi = \mathbf{I} + \frac{\mathbf{F}T}{2!} + \frac{\mathbf{F}^2 T^2}{3!} + \cdots$$

¹³If $w(t)$ is *not* a function of only its sample values, then an integral like that of (2.54) is required to describe its influence on $\mathbf{x}(k+1)$. Random disturbances are treated in Chapter 9.

-
1. Select sampling period T and description matrices \mathbf{F} and \mathbf{G} .
 2. Matrix $\mathbf{I} \leftarrow$ Identity
 3. Matrix $\Psi \leftarrow \mathbf{I}$
 4. $k \leftarrow 11$ [We are using $N = 11$ in (2.60).]
 5. If $k = 1$, go to step 9.
 6. Matrix $\Psi \leftarrow \mathbf{I} + \frac{\mathbf{F}T}{k} \Psi$
 7. $k \leftarrow k - 1$
 8. Go to step 5.
 9. Matrix $\Gamma \leftarrow T\Psi\mathbf{g}$
 10. Matrix $\Phi \leftarrow \mathbf{I} + \mathbf{F}T\Psi$
-

Figure 2.18 Program logic to compute Φ and Γ from \mathbf{F} , \mathbf{G} , and T for simple cases. (The left arrow, \leftarrow , is to be read "is replaced by.")

The Γ integral in (2.56) can be evaluated term by term to give

$$\begin{aligned}
 \Gamma &= \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^{k+1}}{(k+1)!} \mathbf{G} \\
 &= \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^k}{(k+1)!} T\mathbf{G} \\
 &= \Psi T\mathbf{G}.
 \end{aligned} \tag{2.59}$$

We evaluate Ψ by a series in the form

$$\Psi \approx \mathbf{I} + \frac{\mathbf{F}T}{2} \left(\mathbf{I} + \frac{\mathbf{F}T}{3} \left(\cdots \frac{\mathbf{F}T}{N-1} \left(\mathbf{I} + \frac{\mathbf{F}T}{N} \right) \right) \cdots \right), \tag{2.60}$$

which has better numerical properties than the direct series of powers. We then find Γ from (2.59) and Φ from (2.58). A discussion of the selection of N and a technique to compute Ψ for comparatively large T is given by Källström (1973), and a review of various methods is found in a classic paper by Moler and Van Loan (1978). The program logic for computation of Φ and Γ for simple cases is given in Fig. 2.18. All control design packages that we know of contain logic to compute Φ and Γ from the continuous matrices \mathbf{F} , \mathbf{G} , and the sample period T .¹⁴

¹⁴See X-C2D in Table E.1 in Appendix E.

To compare this method of representing the plant with the discrete transfer functions, we can take the z -transform of (2.57) with $w = 0$ and obtain

$$[z\mathbf{I} - \Phi]\mathbf{X}(z) = \Gamma U(z), \quad (2.61a)$$

$$Y(z) = \mathbf{H}\mathbf{X}(z); \quad (2.61b)$$

therefore

$$\frac{Y(z)}{U(z)} = \mathbf{H}[z\mathbf{I} - \Phi]^{-1}\Gamma. \quad (2.62)$$

Example 2.13: For the satellite attitude-control example, the Φ and Γ matrices are easy to calculate using (2.58) and (2.59) and the values for \mathbf{F} and \mathbf{G} defined in (2.45). Since $\mathbf{F}^2 = 0$ in this case, we have

$$\begin{aligned} \Phi &= \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F}^2 T^2}{2!} + \cdots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \\ \Gamma &= \left[\mathbf{I}T + \mathbf{F} \frac{T^2}{2!} + \frac{\mathbf{F}^2 T^3}{3!} \right] \mathbf{G} \\ &= \left\{ \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{T^2}{2} \right\} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}; \end{aligned}$$

hence, using (2.61), we obtain

$$\begin{aligned} \frac{Y(z)}{U(z)} &= [1 \quad 0] \left\{ z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right\}^{-1} \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ &= \frac{T^2}{2} \frac{(z+1)}{(z-1)^2}, \end{aligned}$$

which is the same result that would be obtained using (2.39) and the z -transform tables.

Note that to compute Y/U we find that the denominator is the determinant $\det(z\mathbf{I} - \Phi)$, which comes from the matrix inverse in (2.62). This

determinant is the characteristic polynomial of the transfer function, and the zeros of the determinant are the poles of the plant. We have two poles at $z = 1$ in this case, corresponding to the two integrations in this plant's equations of motion.

We can explore further the question of poles and zeros and the state-space description by considering again the transform equations (2.61). An interpretation of transfer-function poles from the perspective of the corresponding difference equation is that a pole is a value of z such that the equation has a nontrivial solution when the forcing input is zero. From (2.61a), this implies that the linear eigenvalue equations

$$[z\mathbf{I} - \Phi]\mathbf{X}(z) = [0]$$

have a nontrivial solution. From matrix algebra the well-known requirement for this is that $\det(z\mathbf{I} - \Phi) = 0$. In the present case, we have

$$\begin{aligned}\det[z\mathbf{I} - \Phi] &= \det \left[\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right] \\ &= \det \begin{bmatrix} z-1 & T \\ 0 & z-1 \end{bmatrix} \\ &= (z-1)^2 = 0,\end{aligned}$$

which is the characteristic equation, as we have seen. To compute the poles numerically when the matrices are given, one would use an eigenvalue routine.¹⁵

Along the same line of reasoning, a system zero is a value of z such that the system output is zero even with a nonzero state-and-input combination. Thus if we are able to find a nontrivial solution for $\mathbf{X}(z_0)$ and $U(z_0)$ such that $Y(z_0)$ is zero, then z_0 is a zero of the system. Combining the two parts of (2.57), we must satisfy the requirement

$$\begin{bmatrix} z\mathbf{I} - \Phi & -\Gamma \\ \mathbf{H} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(z) \\ U(z) \end{bmatrix} = [0]. \quad (2.63)$$

Once more the condition for the existence of nontrivial solutions is that the determinant of the square coefficient system matrix be zero.¹⁶ For the

¹⁵See EIGENV in Table E.1.

¹⁶We do not consider here the case of different numbers of inputs and outputs.

satellite example, we have

$$\begin{aligned}
 \det \begin{bmatrix} z-1 & -T & -T^2/2 \\ 0 & z-1 & -T \\ 1 & 0 & 0 \end{bmatrix} &= 1 \cdot \det \begin{bmatrix} -T & -T^2/2 \\ z-1 & -T \end{bmatrix} \\
 &= +T^2 + \left(\frac{T^2}{2}\right)(z-1) \\
 &= +\frac{T^2}{2}z + \frac{T^2}{2} \\
 &= +\frac{T^2}{2}(z+1).
 \end{aligned}$$

Thus we have a single zero at $z = -1$, as we have seen from the transfer function. Again, to compute the values of the zeros, called transmission zeros, good algorithms exist in matrix algebra.¹⁷

2.4.4 State-Space Models for Systems with Delay

Thus far we have discussed the calculation of discrete state models from continuous, ordinary differential equations of motion. Now we present the formulas for including a time delay in the model and also a time prediction up to one period which corresponds to the modified z -transform as defined by Jury. We begin with a state-variable model that includes a delay in control action. The state equations are

$$\begin{aligned}
 \dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) + \mathbf{G}u(t - \lambda), \\
 y &= \mathbf{H}\mathbf{x}.
 \end{aligned} \tag{2.64}$$

The general solution to (2.64) is given by (2.53); it is

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{G}u(\tau - \lambda) d\tau.$$

If we let $t_0 = kT$ and $t = kT + T$, then

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+T-\tau)}\mathbf{G}u(\tau - \lambda) d\tau.$$

¹⁷See ZEROS in Table E.1. In using this function, one must be careful to account properly for the zeros that are at infinity; the function might return them as very large numbers that the user must remove to uncover the finite zeros.

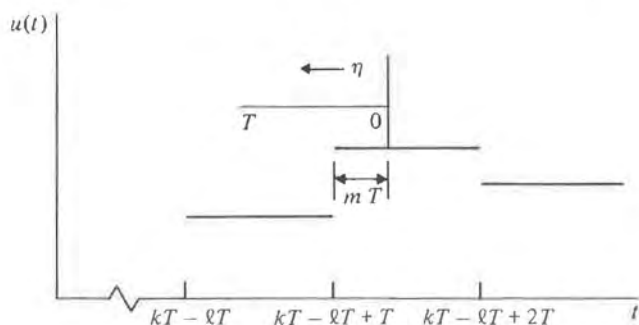


Figure 2.19 Sketch of a piecewise input and time axis for a system with time delay.

If we substitute $\eta = kT + T - \tau$ for τ in the integral, we find a modification of (2.55):

$$\begin{aligned} \mathbf{x}(kT + T) &= e^{\mathbf{F}T} \mathbf{x}(kT) + \int_T^0 e^{\mathbf{F}\eta} \mathbf{G} u(kT + T - \lambda - \eta) (-d\eta) \\ &= e^{\mathbf{F}T} \mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta} \mathbf{G} u(kT + T - \lambda - \eta) d\eta. \end{aligned}$$

If we now separate the system delay λ into an integral number of sampling periods plus a fraction, we can define an integer ℓ and a positive number m less than one such that

$$\lambda = \ell T - mT, \quad (2.65)$$

and

$$\begin{aligned} \ell &\geq 0, \\ 0 &\leq m < 1. \end{aligned}$$

With this substitution, we find that the discrete system is described by

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T} \mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta} \mathbf{G} u(kT + T - \ell T + mT - \eta) d\eta. \quad (2.66)$$

If we sketch a segment of the time axis near $t = kT - \ell T$ (Fig. 2.19), the nature of the integral in (2.66) with respect to the variable η will become clear. The integral runs for η from 0 to T , which corresponds to t from $kT - \ell T + T + mT$ backward to $kT - \ell T + mT$. Over this period, the control, which we assume is piecewise constant, takes on first the value $u(kT - \ell T + T)$

and then the value $u(kT - \ell T)$. Therefore, we can break the integral in (2.66) into two parts as follows:

$$\begin{aligned} \mathbf{x}(kT + T) &= e^{\mathbf{F}T} \mathbf{x}(kT) + \int_0^{mT} e^{\mathbf{F}\eta} \mathbf{G} d\eta u(kT - \ell T + T) \\ &\quad + \int_{mT}^T e^{\mathbf{F}\eta} \mathbf{G} d\eta u(kT - \ell T) \\ &= \Phi \mathbf{x}(kT) + \Gamma_1 u(kT - \ell T) + \Gamma_2 u(kT - \ell T + T). \end{aligned} \quad (2.67)$$

In (2.67) we defined

$$\Phi = e^{\mathbf{F}T}, \quad \Gamma_1 = \int_{mT}^T e^{\mathbf{F}\eta} \mathbf{G} d\eta, \quad \text{and} \quad \Gamma_2 = \int_0^{mT} e^{\mathbf{F}\eta} \mathbf{G} d\eta. \quad (2.68)$$

To complete our analysis it is necessary to express (2.67) in standard state-space form. To do this we must consider separately the cases of $\ell = 0$, $\ell = 1$, and $\ell > 1$.

For $\ell = 0$, $\lambda = -mT$ according to (2.65), which implies not delay but prediction. Because mT is restricted to be less than T , however, the output will not show a sample before $k = 0$, and the discrete system will be causal. The result is that the discrete system computed with $\ell = 0$, $m \neq 0$ will show the response at $t = 0$, which the same system with $\ell = 0$, $m = 0$ would show at $t = mT$. In other words, by taking $\ell = 0$ and $m \neq 0$ we pick up the response values *between* the normal sampling instants. In z -transform theory, the transform of the system with $\ell = 0$, $m \neq 0$ is called the *modified z -transform*.¹⁸ The state-variable form requires that we evaluate the integral in (2.68). To do so we first convert Γ_1 to a form similar to the integral for Γ_2 . From (2.68) we factor out the constant matrix \mathbf{G} to obtain

$$\Gamma_1 = \int_{mT}^T e^{\mathbf{F}\eta} d\eta \mathbf{G}.$$

If we set $\sigma = \eta - mT$ in this integral, we have

$$\begin{aligned} \Gamma_1 &= \int_0^{T-mT} e^{\mathbf{F}(mT+\sigma)} d\sigma \mathbf{G} \\ &= e^{\mathbf{F}mT} \int_0^{T-mT} e^{\mathbf{F}\sigma} d\sigma \mathbf{G}. \end{aligned} \quad (2.6)$$

¹⁸See Jury (1964) or Ogata (1987).

For notational purposes we will define, for any positive nonzero scalar number, a , the two matrices

$$\Phi(a) = e^{Fa}, \quad \Psi(a) = \frac{1}{a} \int_0^a e^{F\sigma} d\sigma. \quad (2.70)$$

In terms of these matrices, we have

$$\begin{aligned} \Gamma_1 &= (T - mT)\Phi(mT)\Psi(T - mT)G, \\ \Gamma_2 &= mT\Psi(mT)G. \end{aligned} \quad (2.71)$$

The definition (2.70) is also useful from a computational point of view. If we recall the series definition of the matrix exponential,

$$\Phi(a) = e^{Fa} = \sum_{k=0}^{\infty} \frac{F^k a^k}{k!},$$

then we get

$$\begin{aligned} \Psi(a) &= \frac{1}{a} \int_0^a \sum_{k=0}^{\infty} \frac{F^k \sigma^k}{k!} d\sigma \\ &= \frac{1}{a} \sum_{k=0}^{\infty} \frac{F^k}{k!} \frac{a^{k+1}}{k+1} \\ &= \sum_{k=0}^{\infty} \frac{F^k a^k}{(k+1)!}. \end{aligned} \quad (2.72)$$

But now we note that the series for $\Phi(a)$ can be written as

$$\Phi(a) = I + \sum_{k=1}^{\infty} \frac{F^k a^k}{k!}.$$

If we let $k = j + 1$ in the sum, then, as in (2.58), we have

$$\begin{aligned} \Phi(a) &= I + \sum_{j=0}^{\infty} \frac{F^{j+1} a^{j+1}}{(j+1)!} \\ &= I + \sum_{j=0}^{\infty} \frac{F^j a^j}{(j+1)!} aF \\ &= I + a\Psi(a)F. \end{aligned} \quad (2.73)$$

The point of (2.73) is that only the series for $\Psi(a)$ needs to be computed and from this single sum we can compute Φ and Γ .

If we return to the case $\ell = 0$, $m \neq 0$, the discrete state equations are

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma_1 u(k) + \Gamma_2 u(k+1),$$

where Γ_1 and Γ_2 are given by (2.71). In order to put these equations in state-variable form, we must eliminate the term in $u(k+1)$. To do this, we define a new state, $\xi(k) = \mathbf{x}(k) - \Gamma_2 u(k)$. Then the equations are

$$\begin{aligned} \xi(k+1) &= \mathbf{x}(k+1) - \Gamma_2 u(k+1) \\ &= \Phi \mathbf{x}(k) + \Gamma_1 u(k) + \Gamma_2 u(k+1) - \Gamma_2 u(k+1), \\ \xi(k+1) &= \Phi [\xi(k) + \Gamma_2 u(k)] + \Gamma_1 u(k) \\ &= \Phi \xi(k) + (\Phi \Gamma_2 + \Gamma_1) u(k) \\ &= \Phi \xi(k) + \Gamma u(k). \end{aligned} \tag{2.74}$$

The output equation is

$$\begin{aligned} y(k) &= \mathbf{H} \mathbf{x}(k) \\ &= \mathbf{H} [\xi(k) + \Gamma_2 u(k)] \\ &= \mathbf{H} \xi(k) + \mathbf{H} \Gamma_2 u(k) \\ &= \mathbf{H}_d \xi(k) + J_d u(k). \end{aligned} \tag{2.75}$$

Thus for $\ell = 0$, the state equations are given by (2.71), (2.74), and (2.75). Note especially that if $m = 0$, then $\Gamma_2 = 0$, and these equations reduce to the previous model with no delay.

Our next case is $\ell = 1$. From (2.67), the equations are given by

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma_1 u(k-1) + \Gamma_2 u(k).$$

In this case, we must eliminate $u(k-1)$ from the right-hand side, which we do by defining a new state $x_{n+1}(k) = u(k-1)$. We have thus an increased dimension of the state, and the equations are

$$\begin{aligned} \begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \end{bmatrix} + \begin{bmatrix} \Gamma_2 \\ 1 \end{bmatrix} u(k), \\ y(k) &= [\mathbf{H} \quad 0] \begin{bmatrix} \mathbf{x} \\ x_{n+1} \end{bmatrix}. \end{aligned} \tag{2.76}$$

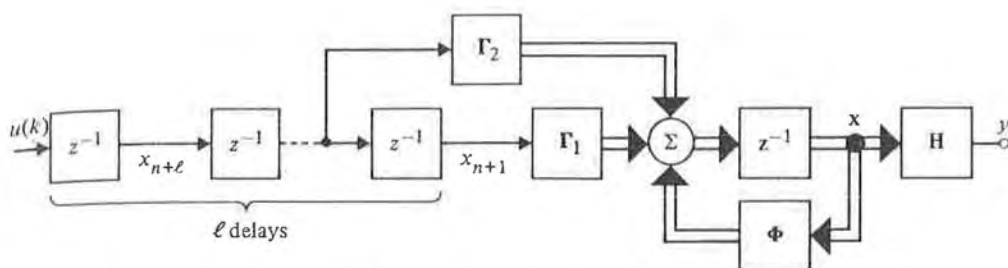


Figure 2.20 Block diagram of system with delay of more than one period. Double line indicates vector valued variables.

For our final case, we consider $\ell > 1$. In this case, the equations are

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma_1 u(k-\ell) + \Gamma_2 u(k-\ell+1),$$

and we must eliminate the past controls up to $u(k)$. To do this we introduce ℓ new variables such that

$$x_{n+1}(k) = u(k-\ell), \quad x_{n+2}(k) = u(k-\ell+1), \dots, x_{n+\ell}(k) = u(k-1).$$

The structure of the equations is

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \\ x_{n+2}(k+1) \\ \vdots \\ x_{n+\ell}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ & & & & 1 & \\ 0 & 0 & 0 & \vdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ x_{n+2}(k) \\ \vdots \\ x_{n+\ell}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k),$$

$$y(k) = [\mathbf{H} \quad 0] \begin{bmatrix} \mathbf{x} \\ x_{n+1} \\ \vdots \\ x_{n+\ell} \end{bmatrix}. \quad (2.77)$$

This final situation is easily visualized in terms of a block diagram, as shown in Fig. 2.20.

2.4.5 Numerical Considerations in Computing Φ and Γ

The numerical considerations of these computations are centered in the approximation to the infinite sum for Ψ given by (2.72) or, for $a = T$, by (2.58).

The problem is that if $\mathbf{F}T$ is large, then $(\mathbf{F}T)^N/N!$ becomes extremely large before it becomes small, and before acceptable accuracy is realized most computer number representations will overflow, destroying the value of the computation. Källström (1973) has analyzed a technique used by Kalman and Englar (1966), which has been found effective by Moler and Van Loan (1978). The basic idea comes from (2.50) with $t_2 - t_0 = 2T$ and $t_1 - t_0 = T$, namely,

$$(e^{\mathbf{F}T})^2 = e^{\mathbf{F}T} e^{\mathbf{F}T} = e^{\mathbf{F}2T}. \quad (2.78)$$

Thus, if T is too large, we can compute the series for $T/2$ and square the result. If $T/2$ is too large, we compute the series for $T/4$, and so on, until we find a k such that $T/2^k$ is *not* too large. We need a test for deciding on the value of k . We propose to approximate the series for Ψ , which can be written

$$\Psi\left(\frac{T}{2^k}\right) = \sum_{j=0}^{N-1} \frac{[\mathbf{F}(T/2^k)]^j}{(j+1)!} + \sum_{j=N}^{\infty} \frac{(\mathbf{F}T/2^k)^j}{(j+1)!} = \hat{\Psi} + \mathbf{R}.$$

We will select k , the factor that decides how much the sample period is divided down, to yield a small remainder term \mathbf{R} . Källström suggests that we estimate the size of \mathbf{R} by the size of the first term ignored in $\hat{\Psi}$, namely,

$$\hat{\mathbf{R}} \cong (\mathbf{F}T)^N / (N+1)! 2^{Nk}.$$

A simpler method is to select k such that the size of $\mathbf{F}T$ divided by 2^k is less than 1. In this case, the series for $\mathbf{F}T/2^k$ will surely converge. The rule is to select k such that

$$2^k > \|\mathbf{F}T\| = \max_j \sum_{i=1}^n |F_{ij}| T.$$

Taking the log of both sides, we find

$$k > \log_2 \|\mathbf{F}T\|,$$

from which we select

$$k = \max(\lceil \log_2 \|\mathbf{F}T\| \rceil, 0), \quad (2.79)$$

where the symbol $\lceil x \rceil$ means the smallest integer greater than x . The maximum of this integer and zero is taken because it is possible that $\|\mathbf{F}T\|$ is

-
-
- | | |
|--|--|
| 1. Select \mathbf{F} and T . | 11. $\Psi \leftarrow I + \frac{\mathbf{F}T_1}{j} \Psi$ |
| 2. Comment: Compute $\ \mathbf{F}T\ $. | 12. $j \leftarrow j - 1$ |
| 3. $V \leftarrow \max_j \{\sum_i F_{ij} \} \times T$ | 13. Go to step 10. |
| 4. $k \leftarrow$ smallest nonnegative integer greater than $\log_2 V$. | 14. Comment: Now double Ψ k times. |
| 5. Comment: compute $\Psi(T/2^k)$. | 15. If $k = 0$, stop. |
| 6. $T_1 \leftarrow T/2^k$ | 16. $\Psi \leftarrow \left(I + \frac{\mathbf{F}T}{2^{k+1}} \Psi \right) \Psi$ |
| 7. $\mathbf{I} \leftarrow$ Identity | 17. $k \leftarrow k - 1$ |
| 8. $\Psi \leftarrow \mathbf{I}$ | go to step 14. |
| 9. $j \leftarrow 11$ | 18. Go to step 15. |
| 10. If $j = 1$, | |
-
-

Figure 2.21 Logic for a program to compute Ψ using automatic time scaling.

already so small that its log is negative, in which case we want to select $k = 0$.

Having selected k , we now have the problem of computing $\hat{\Psi}(T)$ from $\hat{\Psi}(T/2^k)$. Our original concept was based on the series for Φ , which satisfied (2.78). To obtain the suitable formula for Ψ , we use the relation between Φ and Ψ given by (2.58) as follows to obtain the "doubling" formula for Ψ :

$$\begin{aligned}\Phi(2T) &= \Phi(T)\Phi(T), \\ \mathbf{I} + 2T\mathbf{F}\Psi(2T) &= [\mathbf{I} + T\mathbf{F}\Psi(T)][\mathbf{I} + T\mathbf{F}\Psi(T)] \\ &= \mathbf{I} + 2T\mathbf{F}\Psi(T) + T^2\mathbf{F}^2\Psi^2(T); \end{aligned}$$

therefore

$$2T\mathbf{F}\Psi(2T) = 2T\mathbf{F}\Psi(T) + T^2\mathbf{F}^2\Psi^2(T).$$

This is equivalent to

$$\Psi(2T) = \left(\mathbf{I} + \frac{T\mathbf{F}}{2} \Psi(T) \right) \Psi(T),$$

which is the form to be used. The program logic for computing Ψ is

shown in Fig. 2.21.¹⁹ This algorithm does not include the delay discussed in Section 2.4.4. For that, we must implement the logic shown in Fig. 2.20.²⁰

2.5 SIGNAL ANALYSIS AND DYNAMIC RESPONSE

In Section 2.3 we demonstrated that if two variables are related by a linear constant difference equation, then the ratio of the z -transform of the output signal to that of the input is a function of the system equation alone, and the ratio is called the transfer function. A method for study of linear constant discrete systems is thereby indicated, consisting of the following steps:

1. Compute the transfer function of the system $H(z)$.
2. Compute the transform of the input signal, $E(z)$.
3. Form the product, $E(z)H(z)$, which is the transform of the output signal, U .
4. Invert the transform to obtain $u(kT)$.

If the system description is available in difference-equation form, and if the input signal is elementary, then the first three steps of this process require very little effort or computation. The final step, however, is tedious if done by hand, and, because we will later be preoccupied with design of transfer functions to give desirable responses, we attach great benefit to gaining intuition for the kind of response to be expected from a transform without actually inverting it. Our approach to this problem is to present a repertoire of elementary signals with known features and to learn their representation in the transform or z -domain. Thus, when given an unknown transform, we will be able, by reference to these known solutions, to infer the major features of the time-domain signal and thus to determine whether the unknown is of sufficient interest to warrant the effort of detailed time-response computation. To begin this process of attaching a connection between the time domain and the z -transform domain, we compute the transforms of a few elementary signals.

¹⁹See X-C2D in Table E.1

²⁰See DELAY in Table E.1.