3

State Variables and the State Space Description of Dynamic Systems

3.1 INTRODUCTION

In the previous two chapters mathematical models of systems have been presented and discussed using differential or difference equations. In the linear, constant-coefficient case, transfer functions were found to be convenient. In either form, these models were directed toward an input-output description of the system. In this chapter the concept of *state* is introduced and methods of writing state variable forms of the system models are presented. The state variable model of a system includes a description of the internal status of that system, in addition to the input-output behavior. Therefore, state variable models represent a more complete description in general. The state variable approach also applies to time-varying and nonlinear systems which cannot easily be described by transfer functions. Before we begin, a few mathematical conventions must be established.

Functions, Transformations, and Mappings

A *function* is a rule by which elements in one set are associated with elements in another set. A function consists of three things: two specified sets of elements $\mathcal{X} = \{x_i\}$ and $\mathcal{Y} = \{y_i\}$ and a rule relating elements $x_i \in \mathcal{X}$ to elements $y_i \in \mathcal{Y}$. The rule must be unambiguous. That is, for every $x \in \mathcal{X}$, there is associated a single element $y \in \mathcal{Y}$. The rule is often written as

$$y = f(x)$$

The words function, transformation, and mapping will be used interchangeably. A common notation which indicates all three aspects of a function is

$$f: \mathcal{X} \to \mathcal{Y}$$

This states that there is a rule, f, by which every element in $\mathscr{X}$ is mapped into some element in $\mathscr{Y}$. The set $\mathscr{X}$ is called the *domain* of the function, and the set $\mathscr{Y}$ is called the *codomain*. For a particular x, $y = f(x)$ is called the *image* of x, or conversely x is the *pre-image* of y. When the function is applied to every element in $\mathscr{X}$, a set of image points in $\mathscr{Y}$ is generated. This set of images is called the *range* of the function, and is sometimes expressed as $f(\mathscr{X})$.

The words *into* and *onto* are frequently used in conjunction with functions or mappings. The set $f(\mathscr{X})$ is always contained within or possibly equal to $\mathscr{Y}$, written $f(\mathscr{X}) \subseteq \mathscr{Y}$. Thus f is said to map $\mathscr{X}$ *into* $\mathscr{Y}$. If every element in $\mathscr{Y}$ is the image of at least one $x \in \mathscr{X}$, then $f(\mathscr{X}) = \mathscr{Y}$, and the function is said to map $\mathscr{X}$ *onto* $\mathscr{Y}$.

In order that the function be unambiguously defined, there is always just one y associated with each x. However, it is possible that two or more distinct elements $x \in \mathscr{X}$ have the same image point $y \in \mathscr{Y}$. The special case for which distinct elements $x \in \mathscr{X}$ map into distinct elements $y \in \mathscr{Y}$ is called a *one-to-one* mapping. That is, if f is one-to-one, then $x_1 \neq x_2$ implies that $f(x_1) \neq f(x_2)$. Implications such as this are more concisely written as

$$x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$$

As in all logical arguments, negating both propositions reverses the implication, so that f is one-to-one if

$$f(x_1) = f(x_2) \Rightarrow x_1 = x_2$$

If a function is both one-to-one and onto, then for each $y \in \mathscr{Y}$ there is a *unique* pre-image $x \in \mathscr{X}$. The unique relation between y and x defines the inverse function $g = f^{-1}$ with

$$g: \mathscr{Y} \rightarrow \mathscr{X}, \quad \text{where} \quad g[f(x)] = x$$

Vector-Matrix Notation

The bookkeeping conveniences of vector-matrix notation will be used in this chapter. An in-depth and systematic development of these subjects is presented in the following two chapters. A few of the rudimentary notions are now presented. An ordered set of n objects $f_1, f_2, \ldots, f_n$ can be arranged in various array forms, with the position in the array preserving in some way the order of the set. For example, we could write

$$[f_1 \quad f_2 \quad f_3 \quad \cdots \quad f_n] \quad \text{or} \quad \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & \cdots & \end{bmatrix}$$

Which of these forms to use depends on the intended purpose. It is more convenient to refer to the entire array by a single symbol, such as **R**, **C**, or **S**. Ordered arrays of objects (numbers, functions of time, polynomial functions of complex variables, etc.) are called matrices. The first example displayed is a row matrix **R** and the second is a column matrix **C**. **R** and **C** are related by a transpose—i.e., an interchange of rows and columns—written $\mathbf{C}^T = \mathbf{R}$. Either of the first two could be thought of as vectors: a row vector or a column vector. Actually, a vector is more general than just a column of numbers, as is discussed in Chapter 5. The rectangular array **S** in the third representation would normally have its entries designated with two subscripts, a row number i and a column number j. When combining or manipulating arrays in terms of single symbols such as **R, C,** or **S,** a logical algebra must be used if the results are to make sense. The reader is probably familiar with the rules of matrix addition, subtraction, multiplication, and transposition. If not, an occasional reference to Chapter 4 may be useful, since those notions will be used in this chapter.

Vector Space

Let the ordered set of three physical position coordinates (with respect to some coordinate system) be considered as a position vector. Then the set of all such possible vectors can be thought of as a vector space, in this case the three-dimensional physical space. This simple example of a vector space as a set of vectors can be generalized. An ordered n-tuple can be thought of as a point in an n-dimensional space. Certain technical requirements, presented in Chapter 5, are necessary in order to qualify as a valid vector space. Certain relationships (linear functions, mappings, or transformations) between two vectors in finite-dimensional vector spaces can be expressed as a product of a matrix and a vector. The form of the transformation matrix depends upon the coordinate system being used. For example, a point in physical space can be represented in spherical or rectangular coordinates. The column of three numbers that represents this position vector would be very different in the two coordinate systems. The form of the transformation matrix that relates two such vectors would also differ greatly.

3.2 THE CONCEPT OF STATE

The concept of state occupies a central position in modern control theory. However, it appears in many other technical and nontechnical contexts as well. In thermodynamics the equations of *state* are prominently used. Binary sequential networks are normally analyzed in terms of their *states*. In everyday life, monthly financial *state*ments are commonplace. The president's *state* of the Union message is another familiar example.

In all these examples the concept of state is essentially the same. It is a complete summary of the status of the system at a particular point in time. Knowledge of the state at some initial time t_0, plus knowledge of the system inputs after t_0, allows the determination of the state at a later time t_1. As far as the state at t_1 is concerned, it makes no difference how the initial state was attained. Thus the state at t_0 constitutes a complete history of the system behavior prior to t_0, insofar as that history affects future behavior. Knowledge of the present state allows a sharp separation between the past and the future.

At any fixed time the state of a system can be described by the values of a set of variables x_i, called *state variables*. One of the state variables of a thermodynamic system is temperature and its value can range over the continuum of real numbers $\mathcal{R}$. In a binary network state variables can take on only two discrete values, 0 or 1. Note that the state of your checking account at the end of the month can be represented by a single number, the balance. The state of the Union can be represented by such things as gross national product, percent unemployment, the balance of trade deficit, etc. For the systems considered in this book the state variables may take on any scalar value, real or complex. That is, $x_i \in \mathscr{F}$. Although some systems require an infinite number of state variables, only systems which can be described by a finite number n of state variables will be considered here. Then the state can be represented by an n component *state vector* $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$.

The state at a given time belongs to an n-dimensional vector space defined over the field $\mathscr{F}$. A general n-dimensional space will be denoted by $\mathscr{X}^n$. However, because of its great importance, the *state space* will be referred to as Σ from now on.

The systems of interest in this book are dynamic systems. Although a more precise definition is given in the next section, usually the word dynamic refers to something active or changing with time. *Continuous-time* systems have their state defined for all times in some interval, for example, a continually varying temperature or voltage. For *discrete-time* systems the state is defined only at discrete times, as with the monthly financial statement or the annual state of the Union message. Continuous-time and discrete-time systems can be discussed simultaneously by defining the times of interest as $\mathscr{T}$. For continuous-time systems $\mathscr{T}$ consists of the set of all real numbers $t \in [t_0, t_f]$. For discrete-time systems $\mathscr{T}$ consists of a discrete set of times $\{t_0, t_1, t_2, \ldots, t_k, \ldots, t_N\}$. In either case the initial time could be $-\infty$ and the final time could be ∞ in some circumstances.

The state vector $\mathbf{x}(t)$ is defined only for those $t \in \mathscr{T}$. At any given t, it is simply an ordered set of n numbers. However, the character of a system could change with time, causing the *number* of required state variables (and not just the values) to change. If the dimension of the state space varies with time, the notation Σ_t could be used. It is assumed here that Σ is the same n-dimensional state space at all $t \in \mathscr{T}$.

3.3 STATE SPACE REPRESENTATION OF DYNAMIC SYSTEMS

A general class of multivariable control systems is considered. There are r *real*-valued inputs or control variables $u_i(t)$, referred to collectively as the $r \times 1$ vector $\mathbf{u}(t)$. For a fixed time $t \in \mathscr{T}$, $\mathbf{u}(t)$ belongs to the real r-dimensional space $\mathscr{U}^r$. There are m real-valued outputs $y_i(t)$, referred to collectively as the $m \times 1$ vector $\mathbf{y}(t)$. For any $t \in \mathscr{T}$, $\mathbf{y}(t)$ belongs to the real m-dimensional space $\mathscr{Y}^m$. Let the subset of times $t \in \mathscr{T}$ which satisfy $t_a \le t \le t_b$ be denoted as $[t_a, t_b]_{\mathscr{T}}$. For continuous-time systems $[t_a, t_b]_{\mathscr{T}} = [t_a, t_b]$, and for discrete-time systems $[t_a, t_b]_{\mathscr{T}}$ is the intersection of $[t_a, t_b]$ and $\mathscr{T}$.

It is necessary to distinguish between an input function (the graph of $\mathbf{u}(t)$ versus $t \in \mathscr{T}$) and the value of $\mathbf{u}(t)$ at a particular time. The notation $\mathbf{u}_{[t_a, t_b]}$ will be used to indicate a *segment* of an input function over the set of times in $[t_a, t_b]_{\mathscr{T}}$. Similarly, a segment of an output function will be indicated by $\mathbf{y}_{[t_a, t_b]}$. The admissible input functions (actually sequences in the discrete-time case) are elements of the input function space $\mathscr{U}$, that is, $\mathbf{u}$: $\mathscr{T} \to \mathscr{U}$. The output functions (or sequences) are elements of the output function space $\mathscr{Y}$, $\mathbf{y}$: $\mathscr{T} \to \mathscr{Y}$. Heuristically, $\mathscr{U}^r$ and $\mathscr{Y}^m$ can be thought of as "cross sections" through $\mathcal{U}$ and $\mathcal{Y}$ at a particular time t. Similarly, the graph of $\mathbf{x}(t)$ versus t, the so-called *state trajectory,* could be considered as an element of a function space $\mathcal{S}$. The state space Σ can be visualized as a cross section through $\mathcal{S}$ at a particular $t \in \mathcal{T}$.

The primary interest in a control system may be in the relationship between inputs, which can be manipulated, and outputs, which determine whether or not system goals are met. This relationship may be thought of as a mapping or transformation $\mathcal{W}: \mathcal{U} \to \mathcal{Y}$, that is, $\mathbf{y}(t) = \mathcal{W}(\mathbf{u}(t))$. Examples of this type of relationship have been mentioned in Sec. 1.5. It is easy to show that a given $\mathbf{u}_{[t_a, t_b]}$ need *not* define a unique output function $\mathbf{y}_{[t_a, t_b]}$.

EXAMPLE 3.1 The input-output differential equation for the circuit of Figure 3.1 is $dy/dt + (1/RC)y = u(t)/RC$. The solution is

$$y(t) = e^{-(t-t_0)/RC} y(t_0) + (1/RC)\int_{t_0}^{t} e^{-(t-\tau)/RC} u(\tau)\, d\tau$$

A given input function $u(t) \in \mathcal{U}$ defines a *family* of output functions $y(t) \in \mathcal{Y}$. In order to relate one unique output with each input, additional information, such as the value of $y(t_0)$, must be specified. Unless this is done, the nonuniqueness prevents $\mathcal{W}$ from being a transformation in the strict sense. ■

State variables are important because they resolve the nonuniqueness problem illustrated in Example 3.1 and at the same time completely summarize the *internal* status of the system.

Definition 3.1. The state variables of a system consist of a minimum set of parameters which completely summarize the system's status in the following sense. If at any time $t_0 \in \mathcal{T}$, the values of the state variables $x_i(t_0)$ are known, then the output $\mathbf{y}(t_1)$ and the values $x_i(t_1)$ can be *uniquely* determined for any time $t_1 \in \mathcal{T}, t_1 > t_0$, provided $\mathbf{u}_{[t_0, t_1]}$ is known.

Definition 3.2. The state at any time t_0 is a set of the minimum number of parameters $x_i(t_0)$ which allows a *unique* output segment $\mathbf{y}_{[t_0, t]}$ to be associated with each input segment $\mathbf{u}_{[t_0, t]}$ for every $t_0 \in \mathcal{T}$ and for all $t > t_0$, $t \in \mathcal{T}$.

The implications of Definitions 3.1 and 3.2 can be stated in terms of transformations on the input, state, and output spaces. Given (1) a pair of times, t_0 and t_1 in $\mathcal{T}$, (2) $\mathbf{x}(t_0) \in \Sigma$, and (3) a segment $\mathbf{u}_{[t_0, t_1]}$ of an input in $\mathcal{U}$, both the state and the output must be uniquely determinable. This requires that there be a transformation $\mathbf{g}$ which maps the elements $(t_0, t_1, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_1]})$, which can be treated collectively as a single element of

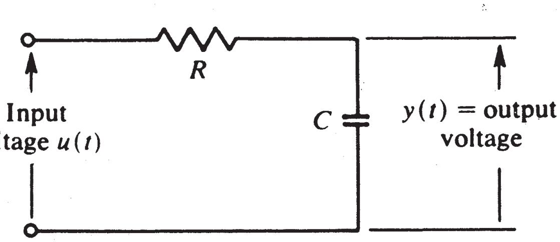


Figure 3.1

the product space (Section 5.10) $\mathcal{T} \times \mathcal{T} \times \Sigma \times \mathcal{U}$, into a unique element in Σ; that is, $\mathbf{g}$: $\mathcal{T} \times \mathcal{T} \times \Sigma \times \mathcal{U} \rightarrow \Sigma$, where

$$\mathbf{x}(t_1) = \mathbf{g}(t_0, t_1, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_1]}) \tag{3.1}$$

Furthermore, since $\mathbf{y}(t_1)$ is uniquely determined, a second transformation exists, $\mathbf{h}$: $\mathcal{T} \times \Sigma \times \mathcal{U}^r \rightarrow \mathcal{Y}^m$ with

$$\mathbf{y}(t_1) = \mathbf{h}(t_1, \mathbf{x}(t_1), \mathbf{u}(t_1)) \tag{3.2}$$

The transformation $\mathbf{h}$ has no memory, rather $\mathbf{y}(t_1)$ depends only on the instantaneous values of $\mathbf{x}(t_1)$, $\mathbf{u}(t_1)$, and t_1. The transformation $\mathbf{g}$ is *nonanticipative* (also called causal). This means that the state, and hence the output at t_1, do not depend on inputs occurring after t_1.

Definition 3.3. The model of a physical system is called a *dynamical system* [1, 2] if a set of times $\mathcal{T}$, spaces $\mathcal{U}$, Σ, and $\mathcal{Y}$, and transformations $\mathbf{g}$ and $\mathbf{h}$ can be associated with it. The transformations are those of Eqs. (*3.1*) and (*3.2*) and must have the following properties:

$$\mathbf{x}(t_0) = \mathbf{g}(t_0, t_0, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_1]}) \quad \text{for any } t_0, t_1 \in \mathcal{T} \tag{3.3}$$

If $\mathbf{u} \in \mathcal{U}$ and $\mathbf{v} \in \mathcal{U}$ with $\mathbf{u} = \mathbf{v}$ over some segment $[t_0, t_1]_{\mathcal{T}}$, then

$$\mathbf{g}(t_0, t_1, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_1]}) = \mathbf{g}(t_0, t_1, \mathbf{x}(t_0), \mathbf{v}_{[t_0, t_1]}) \tag{3.4}$$

If $t_0, t_1, t_2 \in \mathcal{T}$ and $t_0 < t_1 < t_2$, then

$$\begin{aligned} \mathbf{x}(t_2) &= \mathbf{g}(t_0, t_2, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_2]}) \\ &= \mathbf{g}(t_1, t_2, \mathbf{x}(t_1), \mathbf{u}_{[t_1, t_2]}) \\ &= \mathbf{g}(t_1, t_2, \mathbf{g}(t_0, t_1, \mathbf{x}(t_0), \mathbf{u}_{[t_0, t_1]}), \mathbf{u}_{[t_1, t_2]}) \end{aligned} \tag{3.5}$$

Equation (*3.3*) indicates that $\mathbf{g}$ is the identity transformation whenever its two time arguments are the same. Equation (*3.4*), called the *state transition property*, indicates that $\mathbf{x}(t_1)$ does not depend on inputs prior to t_0 except insofar as the past is summarized by $\mathbf{x}(t_0)$. It also indicates that $\mathbf{x}(t_1)$ does not depend on inputs after t_1. Equation (*3.5*), called the *semigroup property*, states that it is immaterial whether $\mathbf{x}(t_2)$ is computed directly from $\mathbf{x}(t_0)$ and $\mathbf{u}_{[t_0, t_2]}$ or if $\mathbf{x}(t_1)$ is first obtained from $\mathbf{x}(t_0)$ and $\mathbf{u}_{[t_0, t_1]}$, and then this state is used, along with $\mathbf{u}_{[t_1, t_2]}$.

Lumped-parameter continuous-time dynamical systems can be represented in state space notation by a set of first-order differential equations (*3.6*) and a set of single-valued algebraic output equations (*3.7*):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{3.6}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \tag{3.7}$$

In order that $\mathbf{x}$ be a valid state vector, Eq. (*3.6*) must have a unique solution. In essence, this means that the components of $\mathbf{f}$ are restricted so that Eq. (*3.8*) defines a unique $\mathbf{x}(t)$ for $t \geq t_0$:

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau \tag{3.8}$$

The theory of differential equations [3] indicates that it is sufficient if $\mathbf{f}$ satisfies a Lipschitz condition with respect to $\mathbf{x}$, is continuous with respect to $\mathbf{u}$, and is piecewise continuous with respect to t. Then there will be a unique $\mathbf{x}(t)$ for any t_0, $\mathbf{x}(t_0)$ provided $\mathbf{u}(t)$ is piecewise continuous. The unique solution of Eq. (*3.6*) defines the transformation $\mathbf{g}$ of Eq. (*3.1*).

EXAMPLE 3.2 Consider a point mass falling in a vacuum. The input is the constant gravitational attraction a and the output is the altitude $y(t)$ shown in Figure 3.2. The input-output relation is $\ddot{y} = -a$. Integrating gives

$$\dot{y}(t) = \dot{y}(t_0) - a(t - t_0)$$

Integrating again gives

$$y(t) = y(t_0) + \dot{y}(t_0)(t - t_0) - a(t - t_0)^2/2$$

The variable y does not constitute the state, since its initial value is not sufficient for uniquely determining $y(t)$. The set of variables y, $\dot{y}$, and $\ddot{y}$ does not form the state, since this is not the *minimum* set of variables required to uniquely determine $y(t)$. The parameter $\ddot{y}$ is not needed. A valid state vector is $\mathbf{x}(t) = [y(t) \quad \dot{y}(t)]^T$. It is easily seen that the form of Eqs. (*3.1*) and (*3.2*) for this example is

$$\mathbf{x}(t) = \begin{bmatrix} 1 & t - t_0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(t_0) - \begin{bmatrix} (t - t_0)^2/2 \\ t - t_0 \end{bmatrix} a, \qquad y(t) = [1 \quad 0]\mathbf{x}(t)$$ ■

Lumped-parameter discrete-time dynamical systems can be described in an analogous way by a set of first-order difference equations (*3.9*) and a set of algebraic output equations (*3.10*):

$$\mathbf{x}(t_{k+1}) = \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k), t_k) \tag{3.9}$$

$$\mathbf{y}(t_k) = \mathbf{h}(\mathbf{x}(t_k), \mathbf{u}(t_k), t_k) \tag{3.10}$$

In general, the functions $\mathbf{f}$ and $\mathbf{h}$ of Eqs. (*3.6*), (*3.7*), (*3.9*), and (*3.10*) can be nonlinear. However, the linear case is of major importance and also lends itself to more detailed analysis. The most general state space representation of a *linear* continuous-time dynamical system is given by Eqs. (*3.11*) and (*3.12*):

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{3.11}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{3.12}$$

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are matrices of dimension $n \times n$, $n \times r$, $m \times n$, and $m \times r$, respectively. Equations (*3.11*) and (*3.12*) are shown in block diagram form in Figure 3.3. The

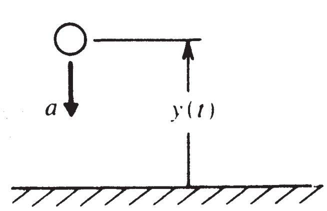


Figure 3.2

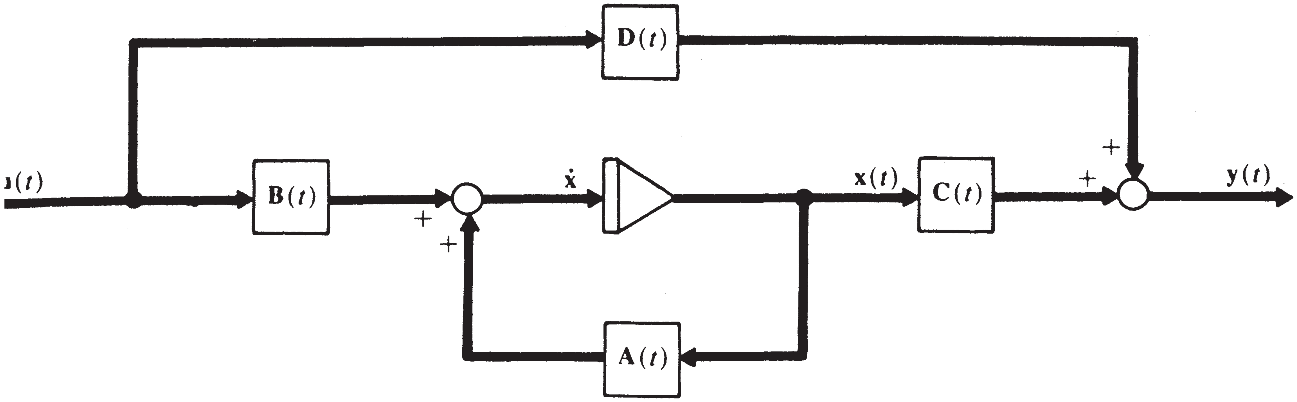


Figure 3.3 State space representation of continuous-time linear system.

heavier lines indicate that the signals are vectors, and the integrator symbol really indicates n scalar integrators.

The state space representation of a discrete-time *linear* system is given by Eqs. (*3.13*) and (*3.14*). The simplified notation k refers to a general time $t_k \in \mathcal{T}$:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \tag{3.13}$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \tag{3.14}$$

The matrices **A, B, C,** and **D** have the same dimensions as in the continuous-time case, but their meanings are different. The block diagram representation of Eqs. (*3.13*) and (*3.14*) is given in Figure 3.4. The delay symbol is analogous to the integrator in Figure 3.3 and really symbolizes n scalar delays.

The notational choice of using the same symbols {**A, B, C, D**} in both cases carries some potential for confusion in certain instances. The advantage of using the same symbols is that many continuous and discrete concepts are revealed as being essentially identical, not only in this chapter but throughout the book. One final word regarding notational symbology seems appropriate. A sizable fraction of the literature uses **F, G,** and **H** in place of **A, B,** and **C** in the continuous case. In the discrete case, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are frequently used in the literature in place of **A** and **B.** It is hoped that with this note of caution the concepts here and in other works can be appreciated and integrated without being tied to a rigid notational standard.

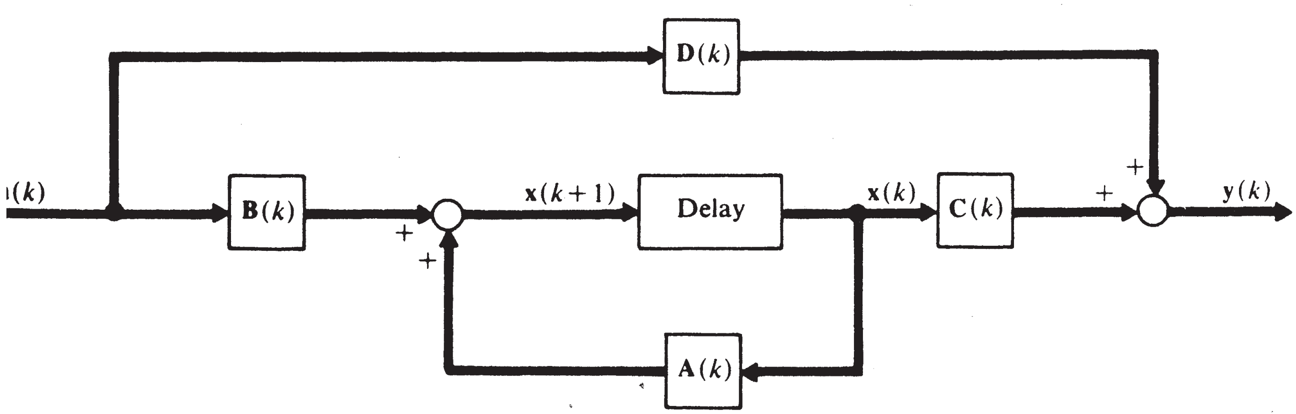


Figure 3.4 State space representation of discrete-time linear system.

Frequently, the discrete-time state variable system models, as in Eqs. (*3.9*) and (*3.10*) or Eqs. (*3.13*) and (*3.14*), are the result of *approximating* a continuous system, or, because of sampling, time-multiplexing of equipment or digital implementation prevents continuous-time operation. Some systems are inherently discrete-time. Other systems are partly discrete and partly continuous, such as a digital controller driving a continuous motor. Some of these sampling and approximation problems will be dealt with in Section 9.8. For present purposes it is assumed that the original system description is in discrete form, perhaps from an empirically obtained ARMA model of Chapter 1 or as a result of taking the Z-transform of a continuous system.

The *analysis* of and *solutions* for linear system state equations are presented in Chapter 9, after the necessary mathematical tools have been developed. The remainder of this chapter develops methods of *obtaining* state space system representations such as those given in Eqs. (*3.6*), (*3.7*) or Eqs. (*3.9*), (*3.10*) for nonlinear systems or Eqs. (*3.11*), (*3.12*) or Eqs. (*3.13*), (*3.14*) for linear systems.

3.4 OBTAINING THE STATE EQUATIONS

3.4.1 From Input-Output Differential or Difference Equations

A class of single-input, single-output systems can be described by an nth-order linear ordinary differential equation:

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_2\frac{d^2 y}{dt^2} + a_1\frac{dy}{dt} + a_0 y = u(t) \tag{3.15}$$

This class of systems can be reduced to the form of n first-order state equations as follows. Define the state variables as

$$x_1 = y, \quad x_2 = \frac{dy}{dt}, \quad x_3 = \frac{d^2 y}{dt^2}, \ldots, \quad x_n = \frac{d^{n-1}y}{dt^{n-1}} \tag{3.16}$$

These particular state variables are often called *phase variables*. As a direct result of this definition, $n-1$ first-order differential equations are $\dot{x}_1 = x_2$, $\dot{x}_2 = x_3, \ldots,$ $\dot{x}_{n-1} = x_n$. The nth equation is $\dot{x}_n = d^n y/dt^n$. Using the original differential equation and the preceding definitions gives

$$\dot{x}_n = -a_0 x_1 - a_1 x_2 - \cdots - a_{n-1}x_n + u(t) \tag{3.17}$$

so that

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \cdots & -a_{n-1} \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) = \mathbf{A}\mathbf{x} + \mathbf{B}u(t) \tag{3.18}$$

The output is $y(t) = x_1(t) = [1 \quad 0 \quad 0 \quad \cdots \quad 0]\,\mathbf{x}(t) = \mathbf{C}\mathbf{x}(t)$. In this case the coefficient matrix $\mathbf{A}$ is the companion matrix (see Section 7.9 and Problem 7.32).

A comparable class of discrete-time systems is described by an nth-order difference equation

$$y(k+n) + a_{n-1}y(k+n-1) + \cdots + a_2 y(k+2) + a_1 y(k+1) + a_0 y(k) = u(k). \tag{3.19}$$

Phase-variable-type states can be defined as

$$x_1(k) = y(k), \quad x_2(k) = y(k+1), \quad x_3(k) = y(k+2), \ldots, \quad x_n(k) = y(k+n-1)$$

where the discrete time points t_k are simply referred to as k. With these definitions, the first $n-1$ state equations are of the form

$$x_i(k+1) = x_{i+1}(k)$$

The original difference equation becomes

$$y(k+n) = x_n(k+1) = -a_0 x_1(k) - a_1 x_2(k) - \cdots - a_{n-1}x_n(k) + u(k) \tag{3.20}$$

Comparison of the continuous and discrete systems just considered shows that they have the identical forms for the **A**, **B**, and **C** system matrices and both have $\mathbf{D} = [0]$. The only difference is that $\mathbf{x}(k+1)$ replaces $\dot{\mathbf{x}}$. In both cases the coefficients a_i could be functions of time, yielding time-variable $\mathbf{A}(t)$ or $\mathbf{A}(k)$ $n \times n$ system matrices.

The fact that the equations have the same *form* should *not* be used to conclude that a discrete approximation to a continuous system can be obtained merely by replacing $\dot{\mathbf{x}}(t)$ with $\mathbf{x}(k+1)$. If Eq. (*3.19*) is an approximation of Eq. (*3.15*), the individual a_i coefficients will be quite different in the two cases.

EXAMPLE 3.3 A continuous-time system is described by

$$\ddot{y} + 4\dot{y} + y = u(t)$$

so that $a_0 = 1$ and $a_1 = 4$. Use the forward difference approximation for derivatives $\dot{y}(t_k) \cong [y(t_{k+1}) - y(t_k)]/T$ and $\ddot{y} \cong [\dot{y}(t_{k+1}) - \dot{y}(t_k)]/T$, where $T = t_{k+1} - t_k$ is the constant sampling period. Find the approximate difference equation.

It follows that $\ddot{y}(t_k) \cong [y(k+2) - 2y(k+1) + y(k)]/T^2$, so that substitution into the differential equation and regrouping terms gives

$$y(k+2) + (4T-2)y(k+1) + (T^2 - 4T + 1)y(k) = T^2 u(k)$$

Thus the discrete coefficients are $a_0 = T^2 - 4T + 1$ and $a_1 = 4T - 2$.

If a backward difference approximation to the derivatives is used, a very different set of coefficients will be found. Also, the time argument on the u input term will change. ■

3.4.2 Simultaneous Differential Equations

The same method of defining the state variables can be applied to multiple-input, multiple-output systems described by several coupled differential equations if the inputs are not differentiated.

EXAMPLE 3.4 A system has three inputs u_1, u_2, u_3 and three outputs y_1, y_2, y_3. The input-output equations are

$$\dddot{y}_1 + a_1\ddot{y}_1 + a_2(\dot{y}_1 + \dot{y}_2) + a_3(y_1 - y_3) = u_1(t)$$

$$\ddot{y}_2 + a_4(\dot{y}_2 - \dot{y}_1 + 2\dot{y}_3) + a_5(y_2 - y_1) = u_2(t)$$

$$\dot{y}_3 + a_6(y_3 - y_1) = u_3(t)$$

Notice that in the second equation $\dot{y}_3$ can be eliminated by using the third equation. State variables are selected as the outputs and their derivatives up to the $(n-1)$th, where n is the order of the highest derivative of a given output.

Select $x_1 = y_1$, $x_2 = \dot{y}_1$, $x_3 = \ddot{y}_1$, $x_4 = y_2$, $x_5 = \dot{y}_2$, $x_6 = y_3$. Then

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_4 = x_5$$

$$\dot{x}_3 = -a_1x_3 - a_2(x_2 + x_5) - a_3(x_1 - x_6) + u_1$$

$$\dot{x}_5 = -a_4(x_5 - x_2 + 2\dot{x}_6) - a_5(x_4 - x_1) + u_2$$

$$\dot{x}_6 = -a_6(x_6 - x_1) + u_3$$

Eliminating $\dot{x}_6$ from the $\dot{x}_5$ equation leads to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -a_3 & -a_2 & -a_1 & 0 & -a_2 & a_3 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ a_5 - 2a_4a_6 & a_4 & 0 & -a_5 & -a_4 & 2a_4a_6 \\ a_6 & 0 & 0 & 0 & 0 & -a_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -2a_4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

The output equation is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}$$

■

When derivatives of the input appear in the system differential equation, the previous method of state variable selection must be modified. If the method is applied without modification, a set of first-order differential equations is obtained as desired, but the input derivatives will still be present. The state equations must express $\dot{\mathbf{x}}$ as a function of $\mathbf{x}$ and $\mathbf{u}$ (and not $\dot{\mathbf{u}}$). A serious mistake that is sometimes made is to define a new vector $\mathbf{u}$ with components made up of u and its derivatives. This is wrong because the inputs to the state equations must be the actual physical inputs to the system. Arbitrary mathematical redefinitions are not allowed on these (or on the output variables $\mathbf{y}$). This differs from the situation for the internal state variables, which may or may not correspond to real physical signals. The input components must be independently selectable control variables. Clearly if $u(t)$ is specified, there is no freedom left in specifying its derivative $\dot{u}(t)$. The correct method of dealing with input derivatives is to somehow absorb the derivative terms into the definitions of the state variables. In simple cases various ad hoc choices may be apparent. For example, consider $\ddot{y} + a\dot{y} + by = u + c\dot{u}$. Rearranging gives $\ddot{y} - c\dot{u} = -a\dot{y} - by + u$. One could use $x_1 = y$, $x_2 = \dot{y} - cu$ so that, assuming c is constant, $\dot{x}_1 = \dot{y} = x_2 + cu$ and $\dot{x}_2 = -a[x_2 + cu] - bx_1 + u$; or

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -b & -a \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c \\ 1 - ac \end{bmatrix} u$$

$$y = [1 \quad 0]\mathbf{x}$$

Since the *selection* of state variables is not a unique process, other choices could be made. For complex higher-order and coupled equations, the ad hoc methods become quite cumbersome. Straightforward systematic methods can be developed, with simulation diagrams being a useful tool.

3.4.3 Using Simulation Diagrams

Equation (*3.8*) indicates that state variables for continuous-time systems are always determined by integrating a function of state variables and inputs. The simulation diagram approach makes use of this fact. Six ideal elements are used as building blocks in the simulation diagrams. They are described in Table 3.1.

Note that a differentiating element $\rightarrow \boxed{d/dt} \rightarrow$ is *not* included. If the equations for a continuous-time system can be simulated using any combination of these elements except the delay, and if no *unnecessary* integrators are used, then the output of each integrator can be selected as a state variable.

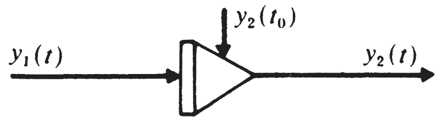
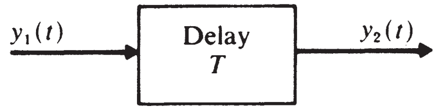
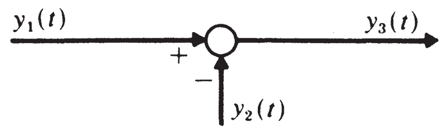
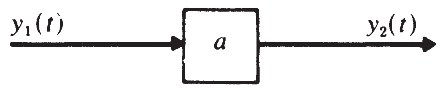
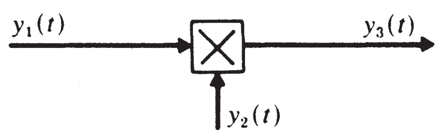
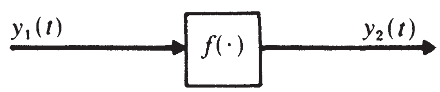
An integrator sums up past inputs to form its present output and hence represents a memory element. For discrete-time systems the ideal delay is used instead of the integrator as the ideal memory element. The close analogy between the integrator and the delay is emphasized in Figure 3.5. The transform domain analogy between $1/s$ and $1/z$ is especially clear. The outputs of delays in a discrete-time simulation diagram constitute a valid choice of state variables. In both cases, the outputs of memory or storage devices constitute states. This notion recurs in Section 3.4.5 where states will be associated with energy storage devices.

EXAMPLE 3.5 One possible simulation diagram for $\ddot{y} + a\dot{y} + by = u + c\dot{u}$ is given in Figure 3.6. Selecting the outputs of the integrators as x_1 and x_2 leads to the same state equations given earlier for this system. ■

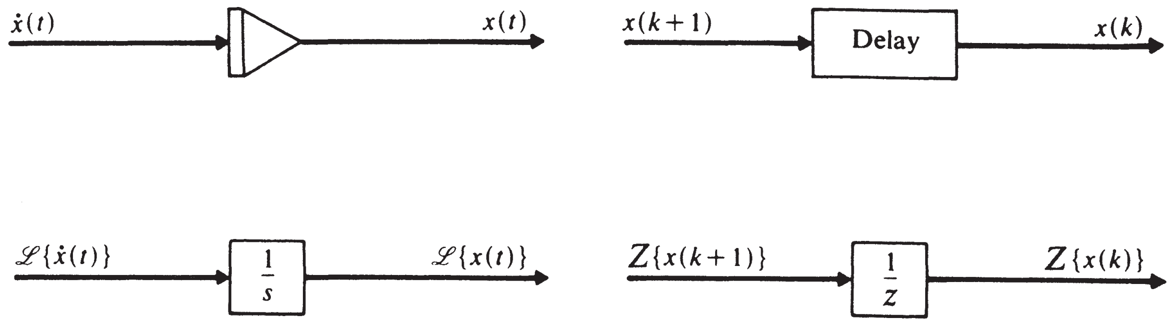
Ad hoc use of the simulation diagram requires a degree of ingenuity in more complicated systems if unnecessary integrators or delays are to be avoided. A systematic approach which is especially simple for constant coefficient systems is now presented. It can be applied with only a modest amount of extra effort to linear, time-variable systems. (See Problems 3.9 and 3.10.) The insight that the procedure provides is also useful in dealing with certain nonlinear systems (see Problem 3.17).

1. Solve each differential equation for its highest derivative. In the case of a difference equation, solve for the most time-advanced term in each equation. A typical nth-order equation is considered for discussion purposes.
2. Formally integrate each differential equation as many times as the highest derivative, here assumed to be n. For difference equations, n delays replace the n-fold integration. In both cases the goal is to achieve the current $y(t)$ or $y(t_k)$ on the left-hand side.

TABLE 3-1

<table>
<tr><th>Element</th><th>Symbol</th><th>Input-output relation</th></tr>
<tr><td>1. Integrator</td><td>
</td><td>$y_2(t) = y_2(t_0) + \int_{t_0}^{t} y_1(\tau)\, d\tau$</td></tr>
<tr><td>2. Delay</td><td>
</td><td>$y_2(t) = y_1(t - T)$</td></tr>
<tr><td>3. Summing junction</td><td>
</td><td>$y_3(t) = y_1(t) - y_2(t)$</td></tr>
<tr><td>4. Gain change</td><td>
</td><td>$y_2(t) = a y_1(t)$</td></tr>
<tr><td>5. Multiplier</td><td>
</td><td>$y_3(t) = y_1(t) y_2(t)$</td></tr>
<tr><td>6. Single-valued function generator</td><td>
</td><td>$y_2(t) = f(y_1(t))$</td></tr>
</table>

3. Group terms on the right-hand side in the form of a nested sequence of integrations (or delays). Use up integrations to remove all derivatives. Use delay operators to remove all advance terms. In constant coefficient cases the coefficients move freely past either of these operators. Time-variable coefficients require use of integration by parts. (See Problems 3.9 and 10.)
4. Draw the simulation diagram by inspection, using the fact that terms inside of an integral sign or delay operator domain constitute inputs to that integrator or delay. A term already containing an integral or delay is the output of another such operator.
5. When the diagram is completed, there should be n integrators or delays. The



(*a*) Ideal integrator (*b*) Ideal delay

Figure 3.5

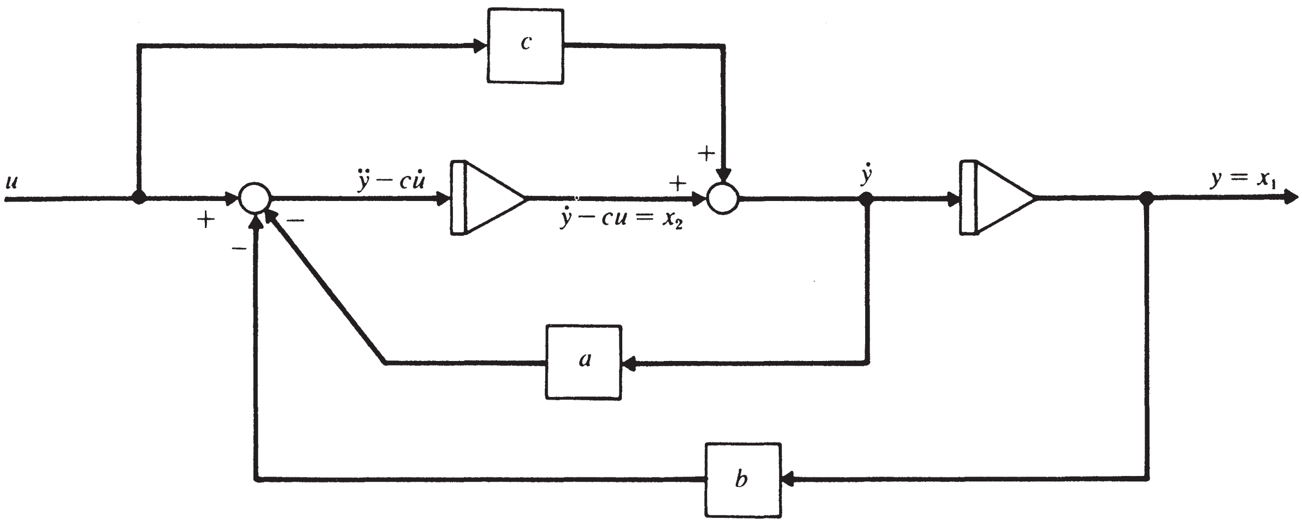


Figure 3.6 ■

output of each integrator is selected as a state $x_i(t)$ in the continuous-time case. The output of each delay in the discrete-time case is selected as a state $x_i(k)$. Then state differential or difference equations can be written directly, using the fact that $\dot{x}_i(t)$ (or $x_i(k+1)$) is the input to that integrator (or delay element).

This procedure leads to state equations in *observable canonical form*. The reason for the name and the importance of the form become apparent in Chapter 11, where properties of state equations are discussed. For now, just note that the **C** output matrix is in an especially simple form, all 0s and 1s, whereas **B** is more complicated. An alternate state variable description is given shortly, in which the input matrix **B** has a similarly simple form, containing all 0s and 1s, but with elements of **C** being more complicated. That form will be called the *controllable canonical form*. Note that the first procedure of Section 3.4, where there were no input derivatives, led to a set of state equations with *both* **B** and **C** having these simple forms.

EXAMPLE 3.6 Consider

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_1\frac{dy}{dt} + a_0 y = \beta_0 u + \beta_1\frac{du}{dt} + \cdots + \beta_m\frac{d^m u}{dt^m}.$$

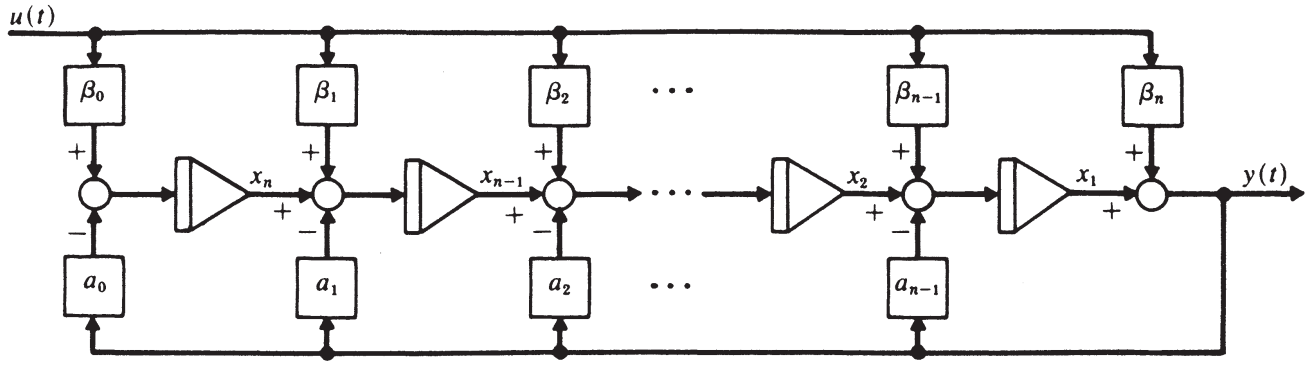


Figure 3.7

The coefficients a_i and β_i are constant. Assume $m = n$. If $m < n$, then some of the β_i terms can be set to zero after the final result is obtained.

(1), (2)
$$y(t) = \underbrace{\int\int\cdots\int}_{n \text{ integrals}} \left\{\beta_n \frac{d^n u}{dt^n} + \left(\beta_{n-1}\frac{d^{n-1}u}{dt^{n-1}} - a_{n-1}\frac{d^{n-1}y}{dt^{n-1}}\right) + \cdots + \left(\beta_1 \frac{du}{dt} - a_1\frac{dy}{dt}\right) + (\beta_0 u - a_0 y)\right\} dt \ldots dt'$$

(3)
$$y(t) = \beta_n u + \int\left\{(\beta_{n-1}u - a_{n-1}y) + \int\left[\beta_{n-2}u - a_{n-2}y + \int\left(\cdots + \int\{\beta_1 u - a_1 y + \int(\beta_0 u - a_0 y)\,dt\}\cdots\right)dt'\right]dt''\right\}dt'''$$

(4) The simulation diagram is shown in Figure 3.7.

(5) Numbering outputs of integrators from the right gives

$$\begin{aligned}
\dot{x}_1 &= -a_{n-1}[x_1 + \beta_n u] + \beta_{n-1}u + x_2 \\
&= -a_{n-1}x_1 + x_2 + (\beta_{n-1} - a_{n-1}\beta_n)u \\
\dot{x}_2 &= -a_{n-2}x_1 + x_3 + (\beta_{n-2} - a_{n-2}\beta_n)u \\
&\vdots \\
\dot{x}_{n-1} &= -a_1 x_1 + x_n + (\beta_1 - a_1\beta_n)u \\
\dot{x}_n &= -a_0 x_1 + (\beta_0 - a_0\beta_n)u
\end{aligned}$$

The output equation is

$$y = x_1 + \beta_n u = [1 \quad 0 \quad 0 \quad \cdots \quad 0]\mathbf{x} + \beta_n u$$

This represents the observable canonical form of the state equations. ■

EXAMPLE 3.7 A system is described by the following equation:

$$y(k+n) + a_{n-1}y(k+n-1) + \cdots + a_1 y(k+1) + a_0 y(k) = \beta_0 u(k) + \beta_1 u(k+1) + \cdots + \beta_m u(k+m)$$

First solve for the most advanced output term (or terms if coupled equations are involved):

$$y(k+n) = -a_{n-1}y(k+n-1) - \cdots - a_1 y(k+1) - a_0 y(k) + \beta_0 u(k) + \beta_1 u(k+1) + \cdots + \beta_m u(k+m)$$

Then delay every term in the equation n times so that $y(k)$ is obtained on the left-hand side. The symbol $\mathcal{D}$ will be used to represent the delay operation. As in Example 3.6, it is assumed that $m = n$ for convenience. If $m < n$, then some of the coefficients β_i can be set to zero. It is impossible that $m > n$ for physically realizable systems:

$$y(k) = -a_{n-1}\mathcal{D}(y(k)) - \cdots - a_1 \mathcal{D}^{n-1}(y(k)) - a_0 \mathcal{D}^n(y(k)) + \beta_0 \mathcal{D}^n(u(k)) + \beta_1 \mathcal{D}^{n-1}(u(k)) + \cdots + \beta_n u(k)$$

Rearrange this expression as a nested sequence of delayed terms:

$$y(k) = \beta_n u(k) + \mathcal{D}\{-a_{n-1}y(k) + \beta_{n-1}u(k) + \mathcal{D}[-a_{n-2}y(k) + \beta_{n-2}u(k) + \mathcal{D}(\cdots + \mathcal{D}\{-a_0 y(k) + \beta_0 u(k)\})]\}$$

The simulation diagram of Figure 3.8 can now be drawn, noting that everything which is operated upon by the delay operator $\mathcal{D}$ forms the input to that delay.

This diagram is exactly like the one of Figure 3.7 except that the integrators have been replaced by delay elements. The state equations can be written down from the simulation diagram by using the fact that if the output of a delay is $x_i(k)$, then its input signal must be $x_i(k+1)$:

$$\begin{aligned}
x_1(k+1) &= -a_{n-1}[x_1(k) + \beta_n u(k)] + x_2(k) + \beta_{n-1}u(k) \\
&= -a_{n-1}x_1(k) + x_2(k) + [\beta_{n-1} - a_{n-1}\beta_n]u(k) \\
x_2(k+1) &= -a_{n-2}x_1(k) + x_3(k) + [\beta_{n-2} - a_{n-2}\beta_n]u(k) \\
&\vdots \\
x_{n-1}(k+1) &= -a_1 x_1(k) + x_n(k) + [\beta_1 - a_1\beta_n]u(k) \\
x_n(k+1) &= -a_0 x_1(k) + [\beta_0 - a_0\beta_n]u(k)
\end{aligned}$$

The output equation is

$$y(k) = x_1(k) + \beta_n u(k) = [1 \quad 0 \quad 0 \quad \cdots \quad 0]\mathbf{x}(k) + \beta_n u(k)$$ ■

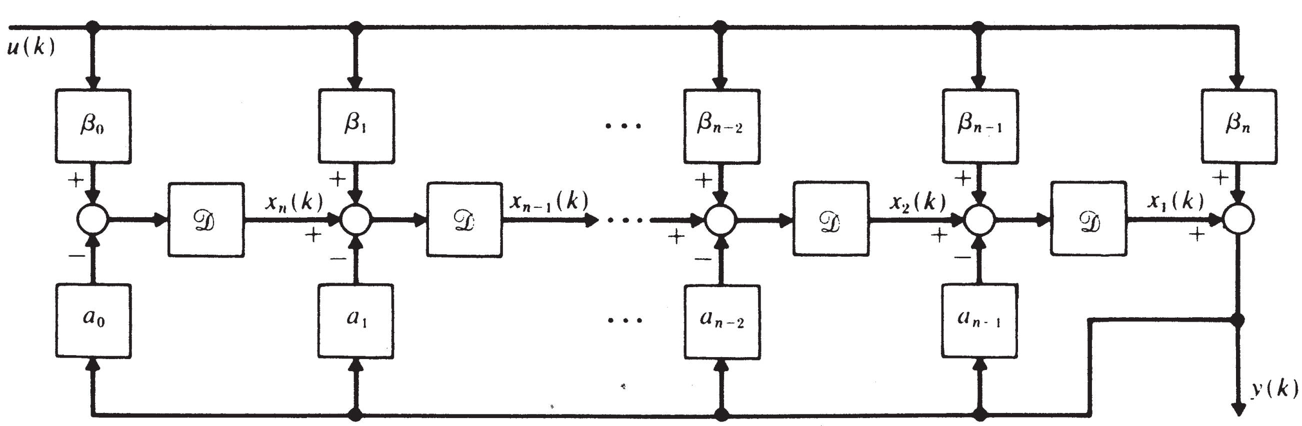


Figure 3.8

3.4.4 State Equations from Transfer Functions

The discussion is restricted to single-input, single-output constant coefficient systems described by a Laplace transform transfer function:

$$\frac{Y(s)}{u(s)} = T(s) = \frac{\beta_m s^m + \beta_{m-1} s^{m-1} + \cdots + \beta_1 s + \beta_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

or a Z-transform transfer function

$$\frac{y(z)}{u(z)} = T(z) = \frac{\beta_m z^m + \beta_{m-1} z^{m-1} + \cdots + \beta_1 z + \beta_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

For any physical system, causality (physical realizability) requires that $m \leq n$. Otherwise, the output $y(k)$ at time t_k would depend upon future inputs $u(j)$ at times t_j with $j > k$. The transfer function can also be written in terms of negative powers of z:

$$T(z) = \frac{z^{m-n}[\beta_m + \beta_{m-1} z^{-1} + \cdots + \beta_1 z^{-(m-1)} + \beta_0 z^{-m}]}{1 + a_{n-1} z^{-1} + a_{n-2} z^{-2} + \cdots + a_1 z^{-(n-1)} + a_0 z^{-n}}$$

Since z^{-1} provides a delay of one sample period, this might be called the delay operator form. It is clear that there is a delay of $n - m$ sample periods from input to output.

The preceding transfer functions correspond exactly to the systems considered in Examples 3.6 and 3.7. Therefore, the previous method of picking states obviously applies. The reason for considering these systems further from the transfer function point of view is that alternative forms of the transfer functions are easily written using algebraic manipulations. The alternative forms can be used to find alternative state variable models. In particular, four major categories of state variable models, called *realizations,* will be presented. Each has its own particular set of advantages, disadvantages, and implications in state variable applications.

1. *Direct realizations* are so named because they derive directly from the expanded polynomial form of the transfer functions. The two major direct realizations are
   (a) Observable canonical form
   (b) Controllable canonical form
2. *Cascade realizations* are so named because they derive from the transfer function written as a product of simple factored terms, which could be represented by a series of cascaded blocks in a block diagram.
3. *Parallel realizations* are so named because they derive from the transfer function written as a sum of partial fraction expansion terms, which would appear as parallel blocks on a block diagram.

Within each of these categories there remains a certain amount of freedom of choice, such as how factors are to be grouped in the cascade form. Actually, the possibilities are infinite because if **x** is any valid state vector, then so is **Sx** for any nonsingular transformation matrix **S.** Finally, there are other canonical realizations not discussed here, such as the lattice and ladder network realizations.

The four major realizations discussed in this book are now illustrated for continuous-time systems. The observable canonical form has already been derived using the nested integrator approach. Its simulation diagram is shown in Figure 3.7. The state equation component equations are written in matrix form as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -a_{n-1} & 1 & 0 & 0 & \dots & 0 \\ -a_{n-2} & 0 & 1 & 1 & \dots & 0 \\ \vdots & & & & & \\ -a_1 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \beta_{n-1} - a_{n-1}\beta_n \\ \beta_{n-2} - a_{n-2}\beta_n \\ \vdots \\ \beta_1 - a_1\beta_n \\ \beta_0 - a_0\beta_n \end{bmatrix} u(t)$$

and

$$y(t) = [1 \quad 0 \quad 0 \quad \dots \quad 0]\mathbf{x}(t) + \beta_n u(t) \tag{3.21}$$

The controllable canonical form is obtained by artificially splitting the transfer function denominator polynomial—call it $a(s)$—and the numerator polynomial—call it $b(s)$—as shown in Figure 3.9. An intermediate variable g has been introduced. Now the transfer function from u to g is exactly the same as the transfer function from u to y for Eq. (*3.15*) considered earlier using phase variables. So once again the states can be selected as $x_1 = g$, $x_2 = \dot{g}, \dots, x_n = \overset{(n-1)}{g}$. As a result the differential equation part of the controllable canonical state equations is given by Eq. (*3.17*). The block from g to the output y in Figure 3.9 indicates that $y(t)$ is a linear combination of g and its derivatives, $y(t) = \beta_0 g + \beta_1 \dot{g} + \beta_2 \ddot{g} + \dots + \beta_m \overset{(m)}{g}$. It is assumed that $m = n$ in order to get the most general result. If m is actually less than n, simply set the higher β_i terms to zero in the results to follow. When the state definitions are used, each g term except the last ($d^n g/dt^n$) is simply replaced by the appropriate state. The nth order derivative of g is now $\dot{x}_n$, and this is a combination of all the states and the a_i coefficients. Regrouping terms gives the state output equation for the controllable canonical form as

$$y(t) = [\beta_0 - a_0\beta_n \quad \beta_1 - a_1\beta_n \quad \beta_2 - a_2\beta_n \quad \cdots \quad \beta_{n-1} - a_{n-1}\beta_n]\mathbf{x}(t) + \beta_n u(t) \tag{3.22}$$

When $m < n$, $\beta_n = 0$ and this takes on a deceptively simple form. In order to illustrate the full generality of the result $m = n$ has been assumed. The simulation diagram for the controllable canonical form is given in Figure 3.10. All the feed-forward β_i terms are from $b(s)$ and all the feedback terms are from $a(s)$.

The nth-order s-domain transfer function can be written in factored form, Eq. (*3.23*). If all poles p_i are distinct and if $m = n$, the partial fraction expansion form of Eq. (*3.24*) can be obtained:

$$T(s) = \frac{\beta_m(s + z_1)(s + z_2)\cdots(s + z_m)}{(s + p_1)(s + p_2)\cdots(s + p_n)} \tag{3.23}$$

$$T(s) = b_0 + \frac{b_1}{s + p_1} + \frac{b_2}{s + p_2} + \dots + \frac{b_n}{s + p_n} \tag{3.24}$$

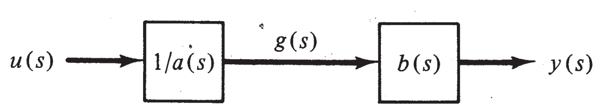


Figure 3.9

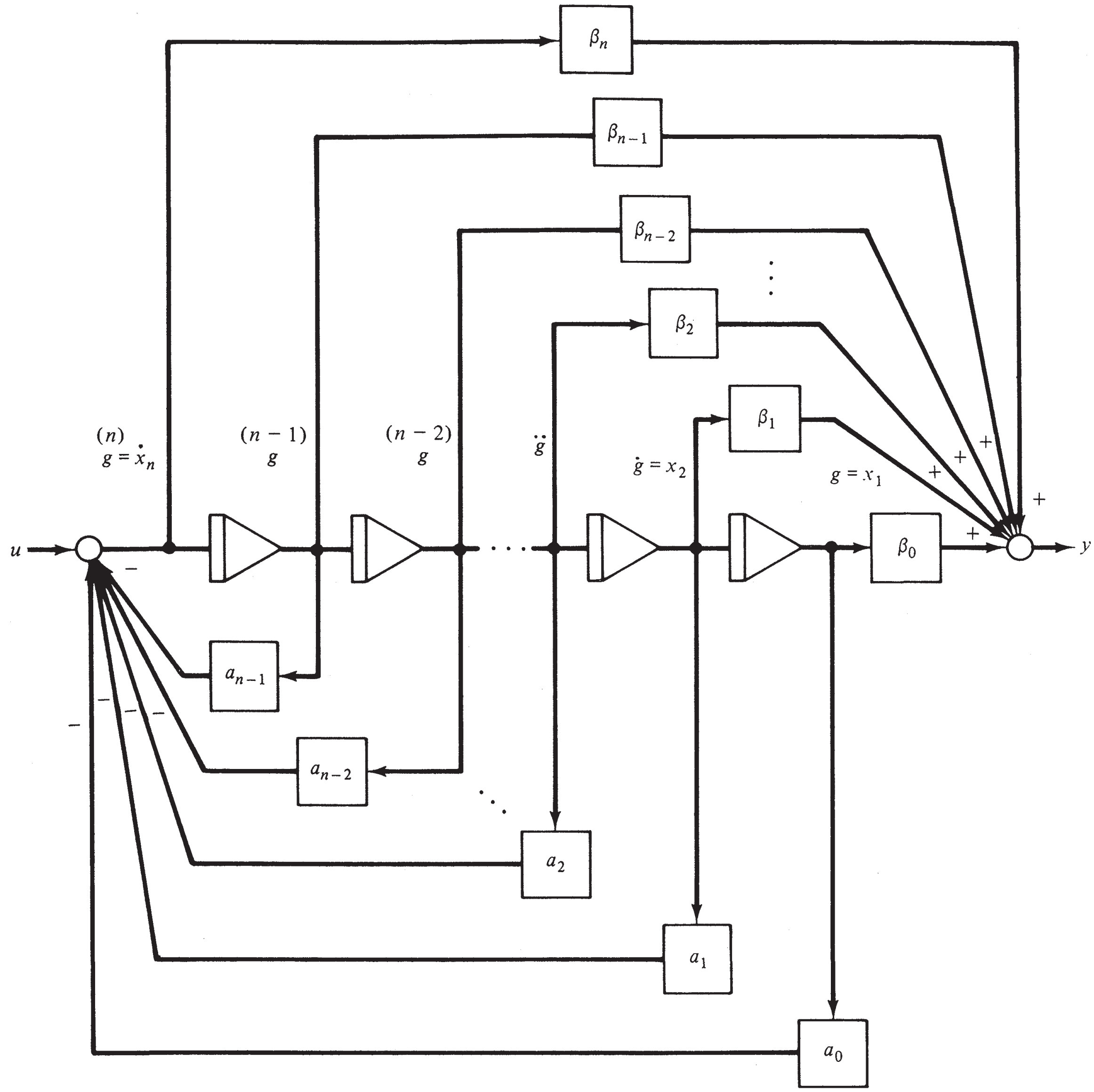


Figure 3.10

If some poles are repeated, the partial fraction expansion will contain additional terms involving powers of the multiple pole $(s + p_i)$ in the denominator (see Problems 3.4 and 3.22). If $m < n$, $b_0 = 0$. For a dynamical system, m can never exceed n.

Equation (3.23) indicates that $T(s)$ can be written as the product of simple factors $1/(s + p_i)$ or $(s + z_j)/(s + p_i)$. Quadratic terms could be considered as well. If a denominator root p_i is complex, it and its conjugate may be kept together in a quadratic factor containing only real coefficients. In any case the system can be represented by a series (cascade) connection of simple first- and/or second-order factors such as those shown in Figure 3.11. The grouping of numerator factors with denominator factors is not discussed here. Each block should be realizable (i.e., the power of the numerator cannot exceed the power of the denominator). Beyond that restriction, how the factors are grouped is left arbitrary, although it does affect the scaling of the signals passed between the blocks. A simulation diagram can be determined for each block by any of

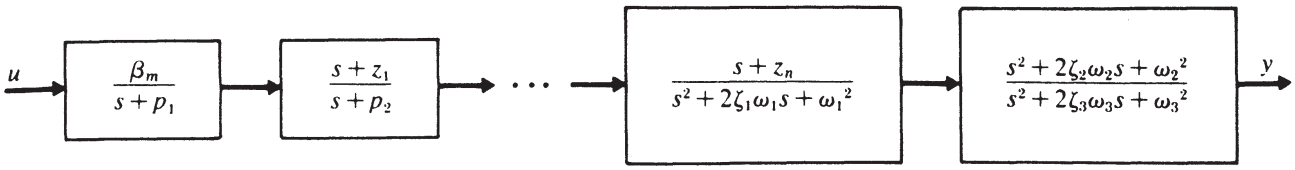


Figure 3.11

the previous methods (see also Problem 3.1). The overall simulation diagram for the cascade realization is the series connection of the diagrams found for each block. The cascade realization of the state equations is then written by selecting integrator outputs as state variables. An example follows shortly.

The partial fraction expanded form in Eq. (*3.24*) indicates that $T(s)$ can be represented as a parallel connection of simple terms (Figure 3.12).

The diagram of Figure 3.12 assumes p_{n-1} is a double pole. The system simulation diagram is also a parallel connection of the individual terms.

EXAMPLE 3.8 Select a suitable set of state variables for the system whose transfer function is

$$T(s) = \frac{s+3}{s^3+9s^2+24s+20}$$

Note that in factored form

$$T(s) = \frac{s+3}{(s+2)^2(s+5)}$$

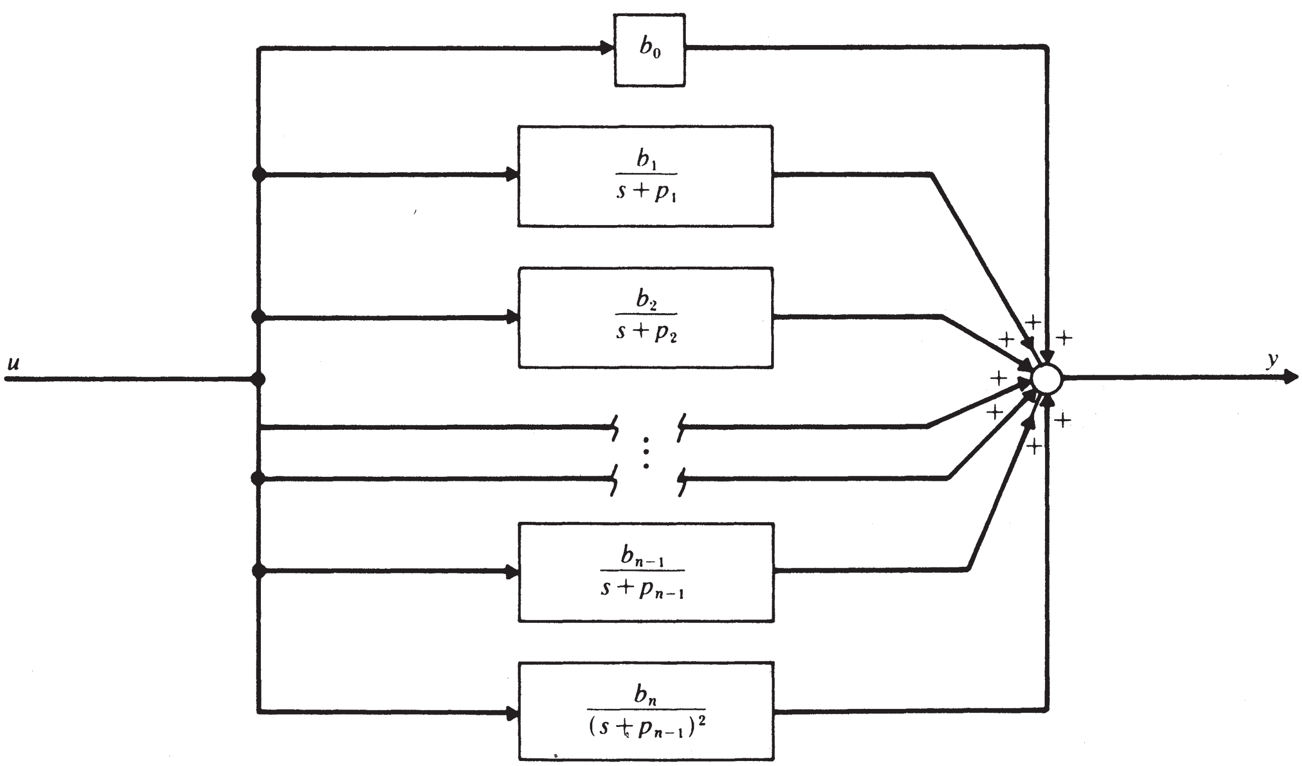


Figure 3.12

and, using a partial fraction expansion,

$$T(s) = \frac{\frac{2}{9}}{s+2} + \frac{\frac{1}{3}}{(s+2)^2} + \frac{-\frac{2}{9}}{s+5}$$

Using the original form of the transfer function will lead to one possible direct realization. First, it is noted that

$$\dddot{y} = -9\ddot{y} - 24\dot{y} - 20y + \dot{u} + 3u$$

or

$$y = \int\left\{-9y + \int\left[-24y + u + \int(-20y + 3u)\,dt\right]dt'\right\}dt''$$

from which the simulation diagram of Figure 3.13 is obtained.

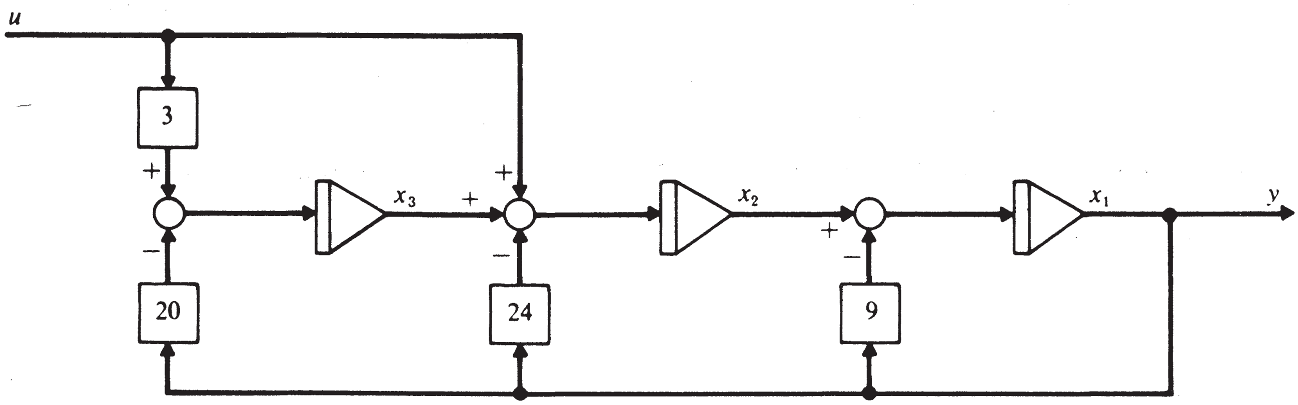


Figure 3.13

Using Figure 3.13, the observable canonical form of the state equations is

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}-9 & 1 & 0\\ -24 & 0 & 1\\ -20 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 1\\ 3\end{bmatrix}u$$

and

$$y = [1 \quad 0 \quad 0]\mathbf{x}$$

This could have been written directly from the general result of Eq. (*3.21*).

A second direct form of the transfer function will now be developed. The numerator and denominator of the transfer function are separated, as shown in Figure 3.14. The intermediate variable thus created is labeled g for convenience. The relationship between u and g is given by

$$\dddot{g} + 9\ddot{g} + 24\dot{g} + 20g = u$$

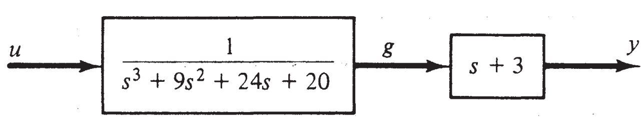


Figure 3.14

This means that the **A** matrix will be in companion form and the **B** matrix will assume its simplest possible form—all 0s or 1s.

The relationship between the fictitious g and the output y depends only on the numerator of $T(s)$. In this simple case

$$y = \dot{g} + 3g$$

Noting that since the s variable indicates differentiation, y is seen to be a linear combination of g and its various derivatives. These derivatives are available as inputs to the various integrators. Using this reasoning leads to the simulation diagram of Figure 3.15. Then, by picking outputs of integrators as states, the controllable canonical form of the state equations is as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -20 & -24 & -9 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [3 \quad 1 \quad 0]\mathbf{x} + [0]u$$

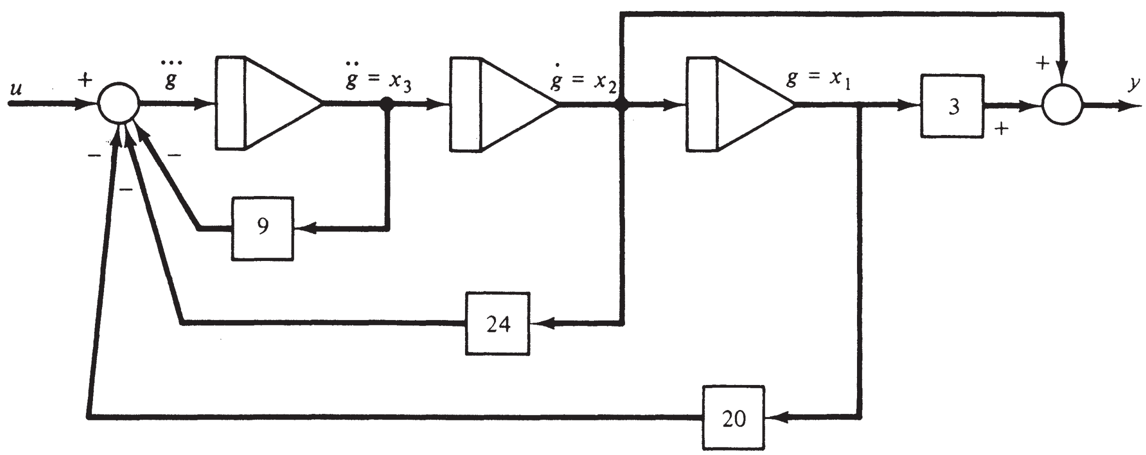


Figure 3.15

These could have been written directly from Eqs. (*3.18*) and (*3.22*).

Using the factored form of $T(s)$, the simulation diagram of Figure 3.16 is obtained. From Figure 3.16, one possible cascade realization is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -5 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} u \quad \text{and} \quad y = [1 \quad 0 \quad 0]\mathbf{x}$$

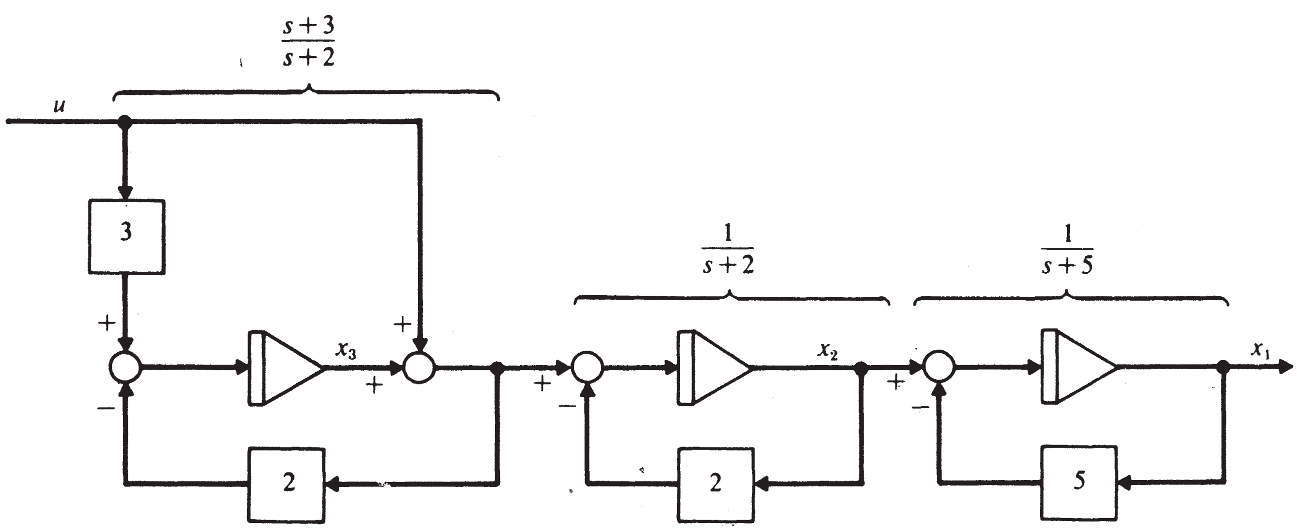


Figure 3.16

Using the partial fraction expansion, the simulation diagram of Figure 3.17 is obtained. From Figure 3.17, the parallel realization is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u \quad \text{and} \quad y = \begin{bmatrix} -\frac{2}{9} & \frac{1}{3} & \frac{2}{9} \end{bmatrix} \mathbf{x}$$

■

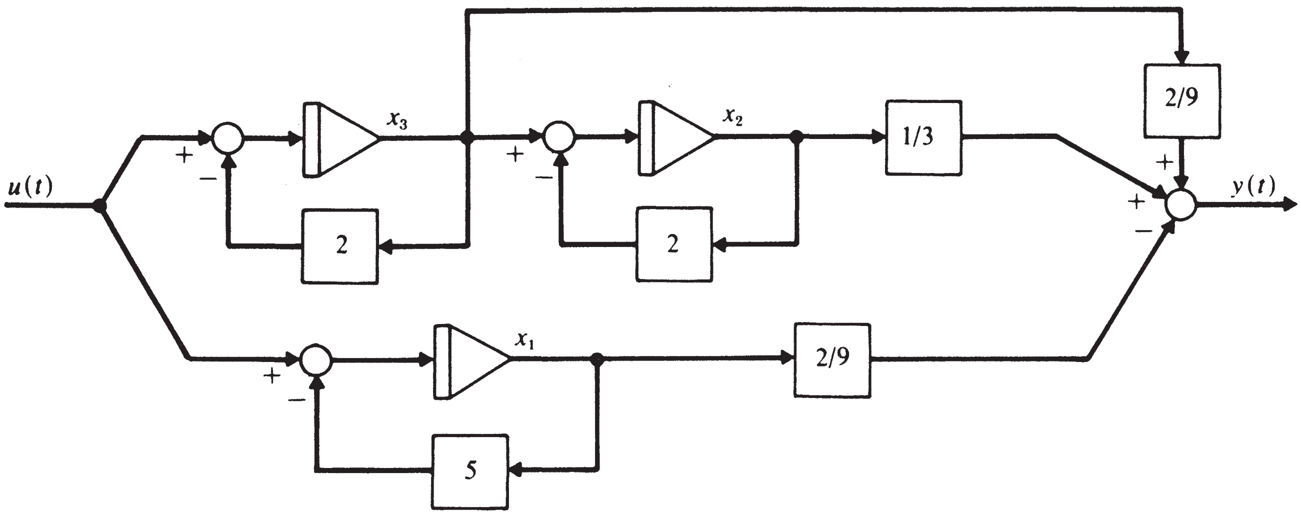


Figure 3.17

Four different sets of valid state equations have been derived for this system. In the third and fourth forms, the diagonal terms of the matrix **A** are the same, the system poles (and, as we see in Chapter 7, the eigenvalues of **A**).The fourth form gives **A** in Jordan canonical form (see Chapter 7). All four representations are different, but all have the same number of state variables, and this is the order of the system. They also all have the same eigenvalues (poles). Note that if separate parallel paths had been used for the terms $1/(s+2)^2$ and $1/(s+2)$, one unnecessary integrator would have been used. This should be avoided, since unnecessary integrators means unnecessary state variables. This relates to the topic of minimal realizations in Chapter 12 and controllability and observability properties of Chapter 11.

The same four realizations can be found for discrete-time systems described by Z-transform transfer functions. Much of the work is exactly the same, with delay operators replacing integrators. Therefore, the major points are presented by way of an example. To give a comparison with the continuous-system results, the third-order system of Example 3.8 is used. When it is preceded by a zero-order hold and then Z-transformed with a sampling period of 0.2, the resulting discrete transfer function is

$$T(z) = \frac{0.013667z^2 + 0.00167z - 0.0050}{z^3 - 1.7085z^2 + 0.9425z - 0.1653} \tag{3.25}$$

EXAMPLE 3.9 Direct Realization, Observable Canonical Form The delay operator form of the transfer function converts immediately to the difference equation considered in Example 3.7. The result of the previous example is a set of observable canonical form state equations

$$\mathbf{x}(k+1) = \begin{bmatrix} -a_{n-1} & 1 & 0 & 0 & \dots & 0 & 0 \\ -a_{n-2} & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & & & & & & \vdots \\ -a_1 & 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \beta_{n-1} - a_{n-1}\beta_n \\ \beta_{n-2} - a_{n-2}\beta_n \\ \vdots \\ \beta_1 - a_1\beta_n \\ \beta_0 - a_0\beta_n \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0]\mathbf{x}(k) + \beta_n u(k)$$

For the specific third-order example, the observable canonical state equations are therefore

$$\mathbf{x}(k+1) = \begin{bmatrix} 1.7085 & 1 & 0 \\ -0.9425 & 0 & 1 \\ 0.1653 & 0 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.01361 \\ 0.00167 \\ -0.0050 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0 \quad 0]\mathbf{x}(k)$$

■

EXAMPLE 3.10 Direct Realization, Controllable Canonical Form The previous transfer function is artificially split into two parts with the fictitious variable $g(k)$ in between, as shown in Figure 3.18. The simulation diagram for determining $g(k)$ is first developed using

$$g(k+3) = 1.7085g(k+2) - 0.9425g(k+1) + 0.1653g(k) + u(k)$$

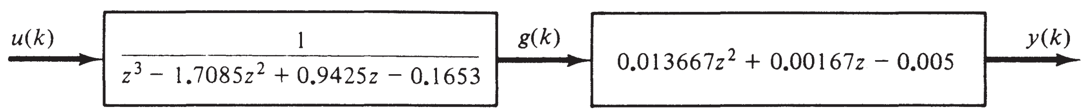


Figure 3.18

Then three successive delay elements give $g(k+2)$, $g(k+1)$, and $g(k)$. This constitutes part of the diagram in Figure 3.19. The second transfer function in Figure 3.18 states that

$$y(k) = -0.0050g(k) + 0.00167g(k+1) + 0.013667g(k+2)$$

This relationship constitutes the rest of Figure 3.19. Numbering the outputs of the delays as states in the order shown immediately gives a controllable canonical form of the state equations

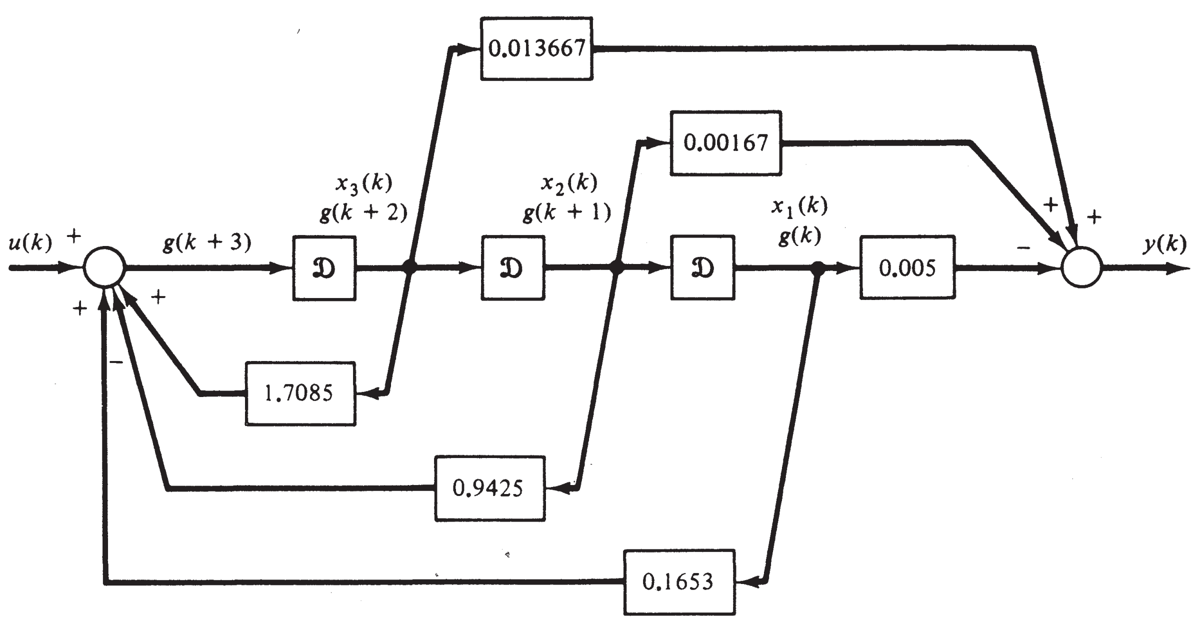


Figure 3.19

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.1653 & -0.9425 & 1.7085 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [-0.0050 \quad 0.00167 \quad 0.013667]\mathbf{x}(k)$$ ■

EXAMPLE 3.11 Cascade Realization The same transfer function is again considered, but this time in factored form (and grouped in the way terms will be cascaded together):

$$T(z) = \left[\frac{1}{z - 0.3679}\right]\left[\frac{z - 0.5488}{z - 0.6703}\right]\left[\frac{z + 0.6714}{z - 0.6703}\right][0.013667]$$

There are at least two valid simulation diagrams for factors such as $\dfrac{z+a}{z-b}$, as shown in Figure 3.20*a* and *b*. Using the second form, the total simulation diagram for a cascade realization is as shown in Figure 3.21. Using state variables numbered as shown, the state equations are

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.6703 & 0.1215 & 1 \\ 0 & 0.6703 & 1 \\ 0 & 0 & 0.3679 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$\begin{aligned} y(k) &= 0.013667[1.3417 \quad 0.1215 \quad 1]\mathbf{x}(k) \\ &= [0.01834 \quad 0.00166 \quad 0.013667]\mathbf{x}(k) \end{aligned}$$ ■

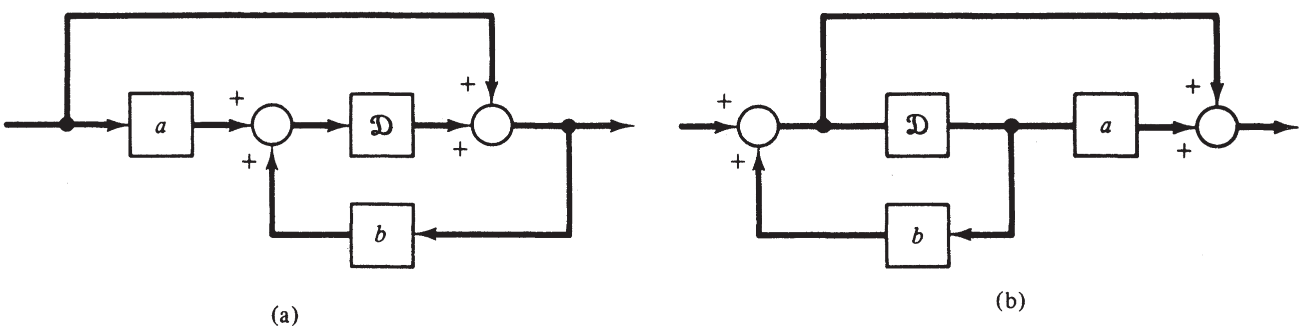


Figure 3.20

EXAMPLE 3.12 Parallel Realization A parallel realization of the same transfer function is now developed using partial fraction expansions. If $T(z)/z$ is expanded and that result is then multiplied by z, the result is

$$T(z) = 0.0305 - 0.07637\frac{z}{(z-0.3679)} + 0.011\frac{z}{(z-0.6703)^2} + 0.0459\frac{z}{(z-0.6703)}$$

Although there is often good reason for going through the extra steps to put an expanded Z-transfer function into this form, there is no reason to do so in the present context. A direct expansion of $T(z)$ in the usual manner gives

$$T(z) = \frac{0.0074}{(z-0.6703)^2} + \frac{0.0418}{(z-0.6703)} - \frac{0.0281}{(z-0.3679)}$$

A simulation diagram of this is shown in Figure 3.22. The state equations are written directly from this as

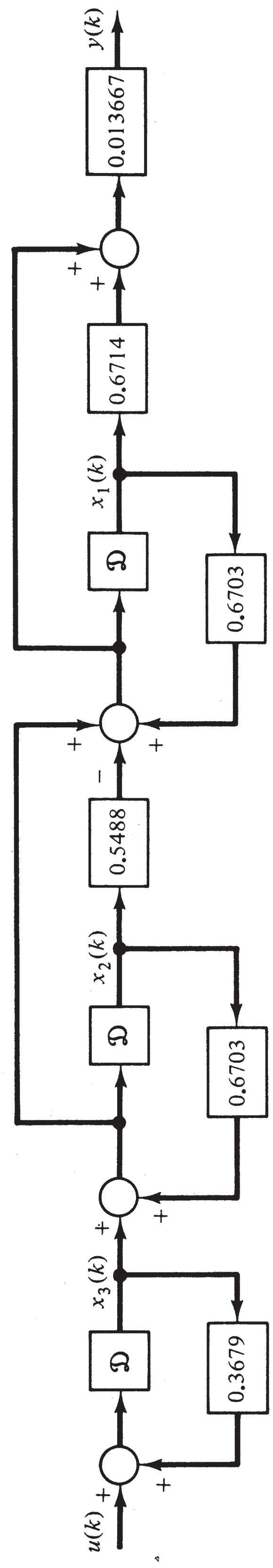


Figure 3.21

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.6703 & 1 & 0 \\ 0 & 0.6703 & 0 \\ 0 & 0 & 0.3679 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [0.0074 \quad 0.0418 \quad -0.0281]\mathbf{x}(k)$$

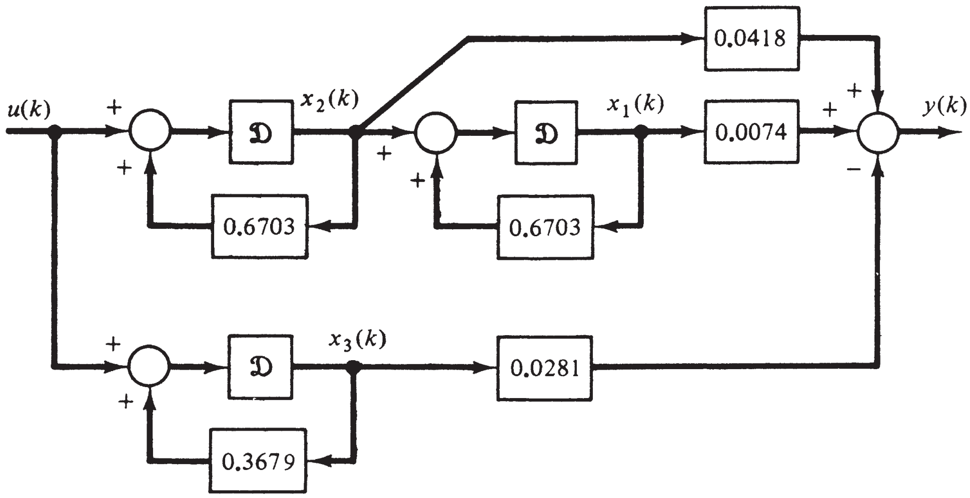


Figure 3.22

As in the continuous-time case, the partial fraction procedure has given an **A** matrix in Jordan canonical form.

The alternative form of the partial fraction expansion equation leads to exactly the same final state equations. It would be a good exercise for the reader to verify this. The *apparent* direct path from u to y through the gain of 0.0305 will cancel out in the manipulations, and a zero D term does in fact result. There is no path from input to output with less than a one sample period delay. ■

3.4.5 State Equations Directly from the System's Linear Graph [4, 5]

Linear graphs were used in Chapter 1 in connection with system modeling. Recall that linear graph techniques are not restricted to linear or constant-coefficient systems. A method of obtaining state equations directly from the system graph is now presented. This is a powerful method because it avoids many of the intermediate manipulations with transfer functions and input-output differential equations, which are restricted to linear, constant systems. Furthermore, the linear graph technique often gives greater engineering insight because the state variables thus obtained are usually related to the energy stored in the system. Before describing the method, a few additional definitions regarding linear graphs are required.

A *tree* is a set of branches of the graph that (1) contains every node of the graph, (2) is connected, and (3) contains no loops. A tree is formed from a graph by removing certain branches. A branch of the graph included in the tree is called a tree branch. Those branches which were deleted while forming a tree are called *links*. Each time a link is added to a tree, one loop is formed. A loop consisting of one link plus a number of tree branches is called the *fundamental loop* associated with that link. For a given tree, if any one tree branch is cut, the tree is separated into two parts. A *fundamental* *cutset* of a given tree branch consists of that one cut tree branch plus all links that connect between nodes of the two halves of the severed tree. In other words, if a line is drawn through the original graph in such a way as to (1) divide the graph into two parts and (2) cut only one tree branch, then that branch plus all links cut by the dividing line form a fundamental cutset.

The following procedure is a systematic method of obtaining state equations from a linear graph:

1. *Form a tree from the graph which includes: (a)* all across variable (voltage) sources; (*b*) as many elements as possible which store energy by virtue of their across variable (capacitors or the analogous elements in other disciplines), that is, elements whose elemental equation has the across variable differentiated; (*c*) elements with algebraic elemental equations (resistors and their analogs); (*d*) as few elements as possible which store energy by virtue of their through variable and have the through variable differentiated in their elemental equation (inductors and their analogs); and (*e*) no through variable (current) sources. (*f*) If ideal transformers are included in the graph, one side of the transformer should be treated like a through variable source and the other side like an across variable source. There will usually be several trees which satisfy these rules.
2. *Choose as state variables the across variables of all capacitor-like elements included in the tree and the through variables of all inductor-like elements not included in the tree.*
3. *The elemental equations for elements involving the selected state variables will be of the form*

   $\dot{x}_i$ = function of through or across variables and inputs

   Use the compatibility laws (Kirchhoff's voltage laws) around the fundamental loops and the conservation laws (Kirchhoff's current laws) into the fundamental cutsets to eliminate nonstate variables from these functional equations.

EXAMPLE 3.13 Consider the linear graph shown in Figure 3.23. The symbols L, C, and R_i are used to indicate the type elements in each branch, although they need not be of an electrical nature.

Using the given rules, the tree of Figure 3.24 is selected. The state variables are the voltage x_1 across C and the current x_2 through L. It is assumed that the elemental equations are nonlinear,

$$C\dot{x}_1 = f_1(i_C) \quad \text{and} \quad L\dot{x}_2 = f_2(v_s - v_1)$$

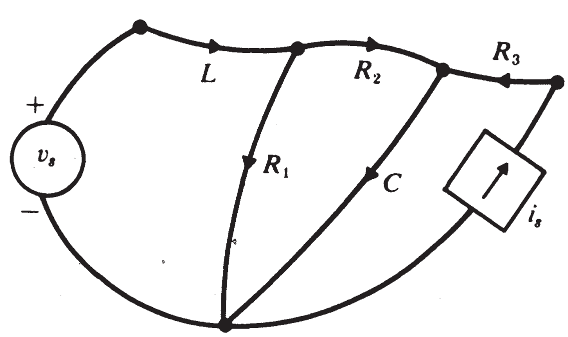


Figure 3.23

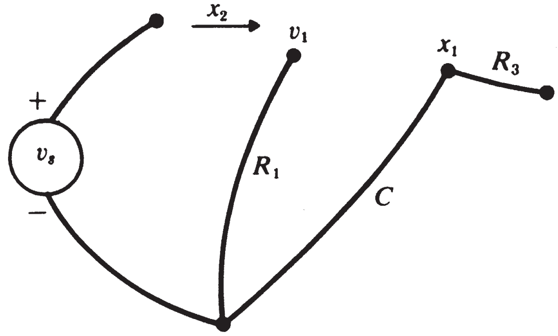


Figure 3.24

To complete the state description, i_C and v_1 must be expressed as functions of x_1, x_2, v_s, and i_s.

The fundamental cutset associated with branch C is given in Figure 3.25 and thus $i_C = i_2 + i_s$. The cutset associated with R_1 is given in Figure 3.26, from which $i_1 = x_2 - i_2$. The fundamental loop formed with link R_2 gives the compatibility equation $v_1 - i_2 R_2 = x_1$, assuming R_2 is a linear resistor. If R_1 is also linear, $v_1 = R_1 i_1$.

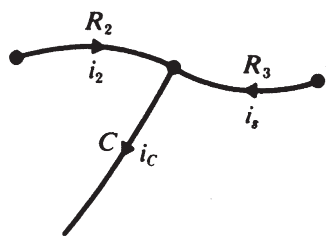


Figure 3.25

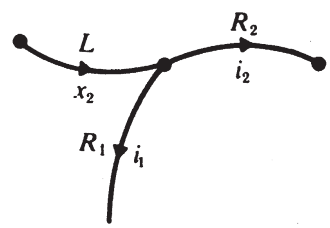


Figure 3.26

These equations are solved simultaneously to give

$$i_C = i_s + \frac{R_1 x_2 - x_1}{R_1 + R_2}$$

$$v_1 = \frac{R_1 R_2 x_2 + R_1 x_1}{R_1 + R_2}$$

so that the state equations are

$$\dot{x}_1 = \frac{1}{C} f_1\left(i_s + \frac{R_1 x_2 - x_1}{R_1 + R_2}\right)$$

$$\dot{x}_2 = \frac{1}{\mathrm{L}} f_2\left(v_s - \frac{R_1 R_2 x_2 + R_1 x_1}{R_1 + R_2}\right)$$

If L and C are also linear, then letting $u_1 = v_s$ and $u_2 = i_s$ reduces the equations to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{C(R_1 + R_2)} & \dfrac{R_1}{C(R_1 + R_2)} \\ -\dfrac{R_1}{L(R_1 + R_2)} & -\dfrac{R_1 R_2}{L(R_1 + R_2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 1/C \\ 1/L & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

If the voltage across R_2 is considered to be the output y, then

$$y = v_1 - x_1 = \left[-\frac{R_2}{R_1 + R_2} \quad \frac{R_1 R_2}{R_1 + R_2} \right] \mathbf{x}$$

■

3.5 INTERCONNECTION OF SUBSYSTEMS

Suppose that two subsystems i and j have been modeled in state variable format:

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i \qquad \dot{\mathbf{x}}_j = \mathbf{A}_j \mathbf{x}_j + \mathbf{B}_j \mathbf{u}_j$$

$$\mathbf{y}_i = \mathbf{C}_i \mathbf{x}_i + \mathbf{D}_i \mathbf{u}_i \qquad \mathbf{y}_j = \mathbf{C}_j \mathbf{x}_j + \mathbf{D}_j \mathbf{u}_j$$

Suppose that the input to subsystem j is the output from subsystem i—that is, $\mathbf{u}_j = \mathbf{y}_i$—and assume that $\mathbf{y}_i$ and $\mathbf{y}_j$ are both considered outputs of the composite system (see Fig. 3.27). The first subsystem's state equations remain unchanged, and by substitution the second set can be written as

$$\dot{\mathbf{x}}_j = \mathbf{A}_j \mathbf{x}_j + \mathbf{B}_j[\mathbf{C}_i \mathbf{x}_i + \mathbf{D}_i \mathbf{u}_i]$$

$$\mathbf{y}_j = \mathbf{C}_j \mathbf{x}_j + \mathbf{D}_j[\mathbf{C}_i \mathbf{x}_i + \mathbf{D}_i \mathbf{u}_i]$$

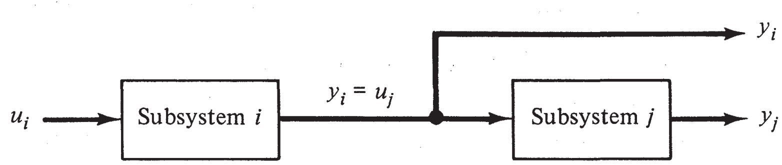


Figure 3.27

The composite system state vector and output vector are

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix}$$

and they satisfy

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ \mathbf{B}_j \mathbf{C}_i & \mathbf{A}_j \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B}_j \mathbf{D}_i \end{bmatrix} \mathbf{u}_i$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C}_i & \mathbf{0} \\ \mathbf{D}_j \mathbf{C}_i & \mathbf{C}_j \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{D}_i \\ \mathbf{D}_j \mathbf{D}_i \end{bmatrix} \mathbf{u}_i$$

Notice that the upper right partitions of the composite system matrices **A** and **C** are both zero. This is because there is no path from subsystem j into i. This substitution approach can be extended to various interconnection topologies. Consider the four-subsystem arrangement of Figure 3.28. The overall system has two groups of inputs and four groups of outputs:

$$\mathbf{u} = [\mathbf{u}_1^T \quad \mathbf{u}_3^T]^T \quad \text{and} \quad \mathbf{y} = [\mathbf{y}_1^T \quad \mathbf{y}_2^T \quad \mathbf{y}_3^T \quad \mathbf{y}_4^T]^T$$

The state vector $\mathbf{x}$ is also the "stacked-up" composite of $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$. After tedious substitution and rearrangement, it is found that the composite system can be described by the state equations

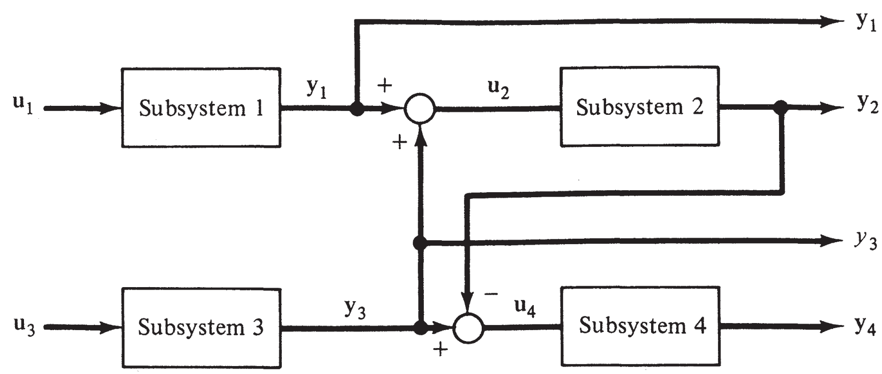


Figure 3.28

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_3 \\ \dot{\mathbf{x}}_4 \end{bmatrix} = \left[\begin{array}{c|c|c|c} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 & \mathbf{B}_2\mathbf{C}_3 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{0} \\ \hline -\mathbf{B}_4\mathbf{D}_2\mathbf{C}_1 & -\mathbf{B}_4\mathbf{C}_2 & \mathbf{B}_4\mathbf{C}_3 - \mathbf{B}_4\mathbf{D}_2\mathbf{C}_3 & \mathbf{A}_4 \end{array}\right] \mathbf{x} + \left[\begin{array}{c|c} \mathbf{B}_1 & \mathbf{0} \\ \hline \mathbf{B}_2\mathbf{D}_1 & \mathbf{B}_2\mathbf{D}_3 \\ \hline \mathbf{0} & \mathbf{B}_3 \\ \hline -\mathbf{B}_4\mathbf{D}_2\mathbf{D}_1 & \mathbf{B}_4\mathbf{D}_3 - \mathbf{B}_4\mathbf{D}_2\mathbf{D}_3 \end{array}\right] \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_3 \end{bmatrix}$$

and

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix} = \left[\begin{array}{c|c|c|c} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{D}_2\mathbf{C}_1 & \mathbf{C}_2 & \mathbf{D}_2\mathbf{C}_3 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{C}_3 & \mathbf{0} \\ \hline -\mathbf{D}_4\mathbf{D}_2\mathbf{C}_1 & -\mathbf{D}_4\mathbf{C}_2 & \mathbf{D}_4\mathbf{C}_3 - \mathbf{D}_4\mathbf{D}_2\mathbf{C}_3 & \mathbf{C}_4 \end{array}\right] \mathbf{x} + \left[\begin{array}{c|c} \mathbf{D}_1 & \mathbf{0} \\ \hline \mathbf{D}_2\mathbf{D}_1 & \mathbf{D}_2\mathbf{D}_3 \\ \hline \mathbf{0} & \mathbf{D}_3 \\ \hline -\mathbf{D}_4\mathbf{D}_2\mathbf{D}_1 & \mathbf{D}_4\mathbf{D}_3 - \mathbf{D}_4\mathbf{D}_2\mathbf{D}_3 \end{array}\right] \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_3 \end{bmatrix}$$

There is obviously a pattern in these matrices that derives from the subsystem interconnection topology, but this is not pursued here. Also note the absence of closed-feedback loops in both these examples. These and other more complicated interconnections are best left until after a more thorough presentation of matrix algebra, including inversion, is presented in the next chapter. Finally, the interconnection procedure just presented can yield a nonminimal state realization in some cases. One such case is when poles and zeros of cascaded blocks cancel. The presumption has been made here that each subsystem represents a real, physical system whose modes and signals are to be preserved in the final model. If doing this results in extra states, so be it. The implications of this shall be made clear in Chapter 11.

3.6 COMMENTS ON THE STATE SPACE REPRESENTATION

The selection of state variables is not a unique process. Various sets of state variables can be used. Some are easier to derive, whereas others are easier to work with once they are obtained. These comments all relate to *mathematics.* Some *physical* considerations also exist. It may be that the starting information about a system is derived from an experimentally obtained transfer function, perhaps by fitting straight-line approximations to a frequency response plot (Bode plot). It could be that an ARMA model has been constructed by fitting to historical data. The procedure you follow in a real situation depends upon what data you have at the start. There is often good reason to select states which have physical significance. These can then at least potentially be measured, perhaps by adding additional instrumentation.

The state equations consist of two generic parts. The differential or difference equation represents the so-called dynamics of the system, and the *algebraic* output equation is often referred to as the output or measurement equation. This can be misleading, since most real sensors have their own inherent dynamical responses. For example, a temperature sensor invariably has some time constant which prevents the instantaneous measurement of the temperature state. Whenever these kinds of instrument dynamics are significant, they must be included in the "dynamical" part of the state equations. That is, they will add states, just as in Section 3.5 when subsystems were cascaded together.

At the risk of oversimplification, it can be said that control theory started and flourished using transfer function methods. Then the state variable approach was developed, and for many years it was synonymous with *modern control.* Some of the advantages of the state variable approach are as follows:

1. It provides a convenient, compact notation and allows the application of the powerful vector-matrix theory, which is developed in the next few chapters of this book.

2. The uniform notation for all systems, regardless of order, makes possible a uniform set of solution techniques and computer algorithms. This is in sharp contrast with, for example, phase-plane methods, which give great insight into the behavior of second-order systems.

3. The state space representation is in an ideal format for computer solution, either analog or digital. In fact, the simulation diagrams used here to write the state equations are ideal starting points for system simulation. This is important because computers are invariably needed in the analysis of all but the most trivial systems.

4. The state space approach originally was able to define and explain more completely many system characteristics and attributes. Currently, most of the advantages and insights gained by use of state variable methods in the early years have been found to have counterparts in the new and expanded input-output transfer function methods of multivariable systems. The two approaches are now both being used. They require somewhat different mathematical tools, but they complement each other in various ways. Some of the newer aspects of transfer function methods appear in later chapters, but this book stresses state variable methods and the linear algebra and matrix theory upon which they depend.

REFERENCES

1. Kalman, R. E.: "Mathematical Description of Linear Dynamical Systems," *Jour. Soc. Ind. Appl. Math-Control Series,* Series A, Vol. 1, No. 2, 1963, pp. 152–192.
2. Desoer, C. A.: *Notes for a Second Course on Linear Systems,* Van Nostrand Reinhold, New York, 1970.
3. Coddington, E. A. and N. Levinson: *Theory of Ordinary Differential Equations,* McGraw-Hill, New York, 1955.
4. Chen, C. T.: *Introduction to Linear System Theory,* Holt, Rinehart and Winston, New York, 1970.

5. Martens, H. R. and D. R. Allen: *Introduction to Systems Theory,* Charles E. Merrill Publishing Co., Columbus, Oh., 1969.
6. Zadah, L. A. and C. A. Desoer: *Linear System Theory, The State Space Approach,* McGraw-Hill, New York, 1963.
7. De Russo, P. M., R. J. Roy, and C. M. Close: *State Variables for Engineers,* John Wiley, New York, 1965.

ILLUSTRATIVE PROBLEMS

Linear, Continuous-Time, Constant Coefficients

3.1 Four input-output transfer functions $y(s)/u(s)$ are given. Describe the systems they represent in state variable form:

(a) $1/(s+\alpha)$ **(b)** $(s+\beta)/(s+\alpha)$

(c) $(s+\beta)/(s^2+2\zeta\omega s+\omega^2)$ **(d)** $(s^2+2\zeta_1\omega_1 s+\omega_1^2)/(s^2+2\zeta_2\omega_2 s+\omega_2^2)$

The solutions are obtained by writing the input-output differential equation, drawing the simulation diagram, and selecting the integrator outputs as state variables.

(a) The differential equation is $\dot{y}+\alpha y=u$. This is simulated in Figure 3.29*a,* from which $\dot{x}=-\alpha x+u$ and $y=x$.

(b) The differential equation is $\dot{y}+\alpha y=\beta u+\dot{u}$. This is simulated in Figure 3.29*b,* from which $\dot{x}=-\alpha x+(\beta-\alpha)u$ and $y=x+u$.

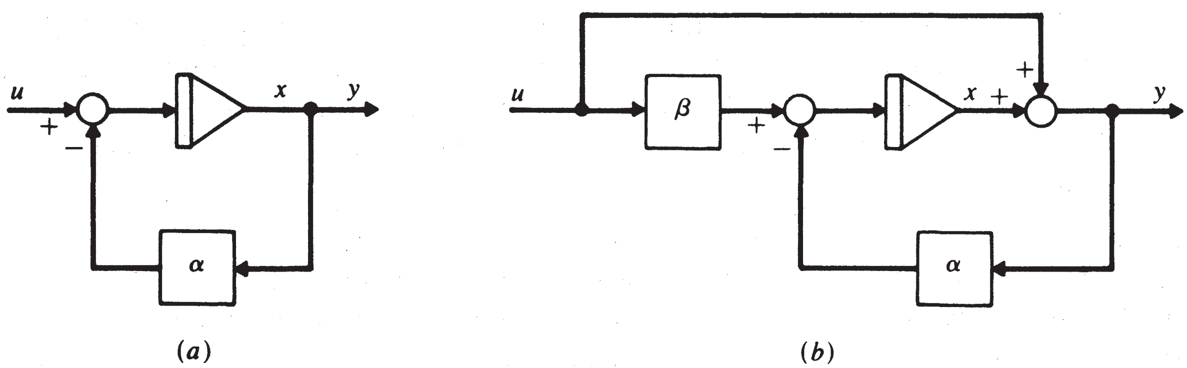


Figure 3.29 (a) and (b)

(c) The differential equation is $\ddot{y}+2\zeta\omega\dot{y}+\omega^2 y=\beta u+\dot{u}$, and is simulated in Figure 3.29*c*. The state equations are $\dot{x}_1=-2\zeta\omega x_1+x_2+u$, $\dot{x}_2=-\omega^2 x_1+\beta u$, and $y=[1\quad 0]\mathbf{x}$.

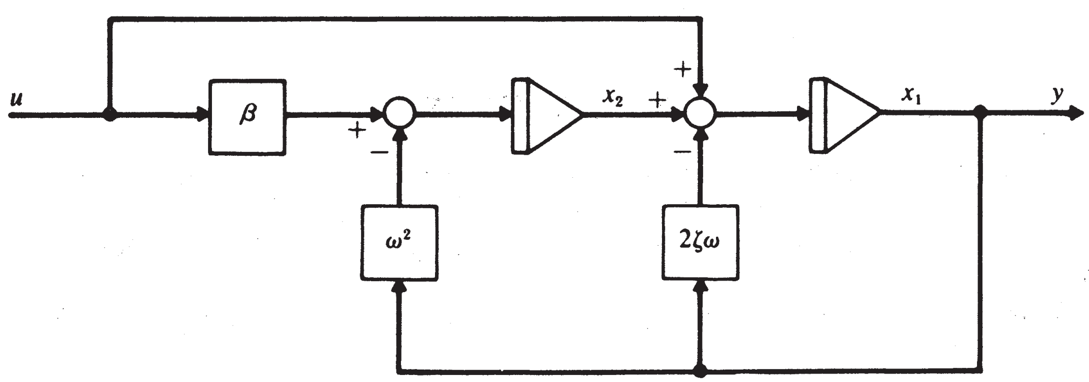


Figure 3.29 (c)

(d) The differential equation is $\ddot{y} + 2\zeta_2\omega_2\dot{y} + \omega_2^2 y = \ddot{u} + 2\zeta_1\omega_1\dot{u} + \omega_1^2 u$. Integrating twice allows this to be written as

$$y = u + \int\left\{2\zeta_1\omega_1 u - 2\zeta_2\omega_2 y + \int[\omega_1^2 u - \omega_2^2 y]\,dt\right\}dt'$$

The simulation diagram of Figure 3.29*d* gives $\dot{x}_1 = -2\zeta_2\omega_2 x_1 + x_2 + (2\zeta_1\omega_1 - 2\zeta_2\omega_2)u$, $\dot{x}_2 = -\omega_2^2 x_1 + (\omega_1^2 - \omega_2^2)u$, and $y = [1 \quad 0]\mathbf{x} + u$.

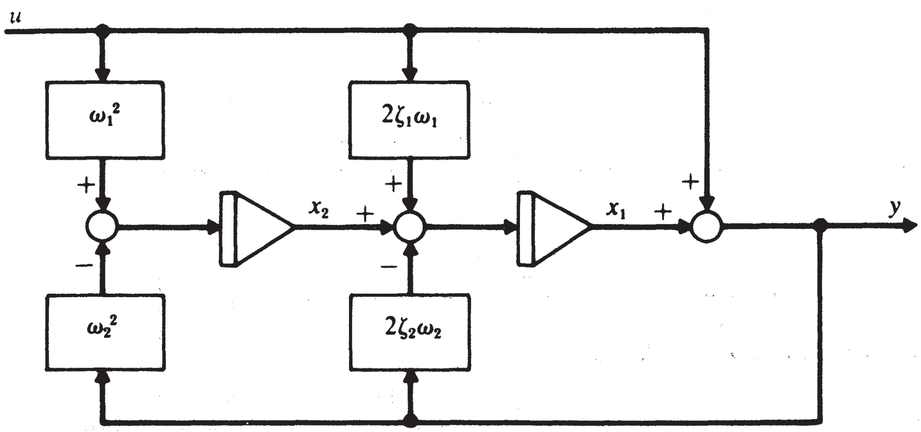


Figure 3.29 (d)

3.2 A system with two inputs and two outputs is described by

$$\ddot{y}_1 + 3\dot{y}_1 + 2y_2 = u_1 + 2u_2 + 2\dot{u}_2 \quad\text{and}\quad \ddot{y}_2 + 4\dot{y}_1 + 3y_2 = \ddot{u}_2 + 3\dot{u}_2 + u_1$$

Select a set of state variables and find the state space equations for this system.

Integrating each equation twice gives

$$y_1 = \iint\{-3\dot{y}_1 - 2y_2 + u_1 + 2\dot{u}_2 + 2u_2\}\,dt\,dt'$$

$$y_2 = \iint\{-4\dot{y}_1 - 3y_2 + \ddot{u}_2 + 3\dot{u}_2 + u_1\}\,dt\,dt'$$

or

$$y_1 = \int\left\{-3y_1 + 2u_2 + \int[-2y_2 + u_1 + 2u_2]\,dt\right\}dt'$$

$$y_2 = u_2 + \int\left\{-4y_1 + 3u_2 + \int[-3y_2 + u_1]\,dt\right\}dt'$$

The simulation diagram of Figure 3.30 can now be drawn.

The state equations are

$$\left.\begin{aligned}\dot{x}_1 &= -3x_1 + x_2 + 2u_2\\ \dot{x}_2 &= -2x_3 + u_1\\ \dot{x}_3 &= -4x_1 + x_4 + 3u_2\\ \dot{x}_4 &= -3x_3 + u_1 - 3u_2\end{aligned}\right\} \quad\text{or}\quad \dot{\mathbf{x}} = \begin{bmatrix}-3 & 1 & 0 & 0\\ 0 & 0 & -2 & 0\\ -4 & 0 & 0 & 1\\ 0 & 0 & -3 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}0 & 2\\ 1 & 0\\ 0 & 3\\ 1 & -3\end{bmatrix}\mathbf{u}$$

and

$$\mathbf{y} = \begin{bmatrix}y_1\\ y_2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\mathbf{u}.$$

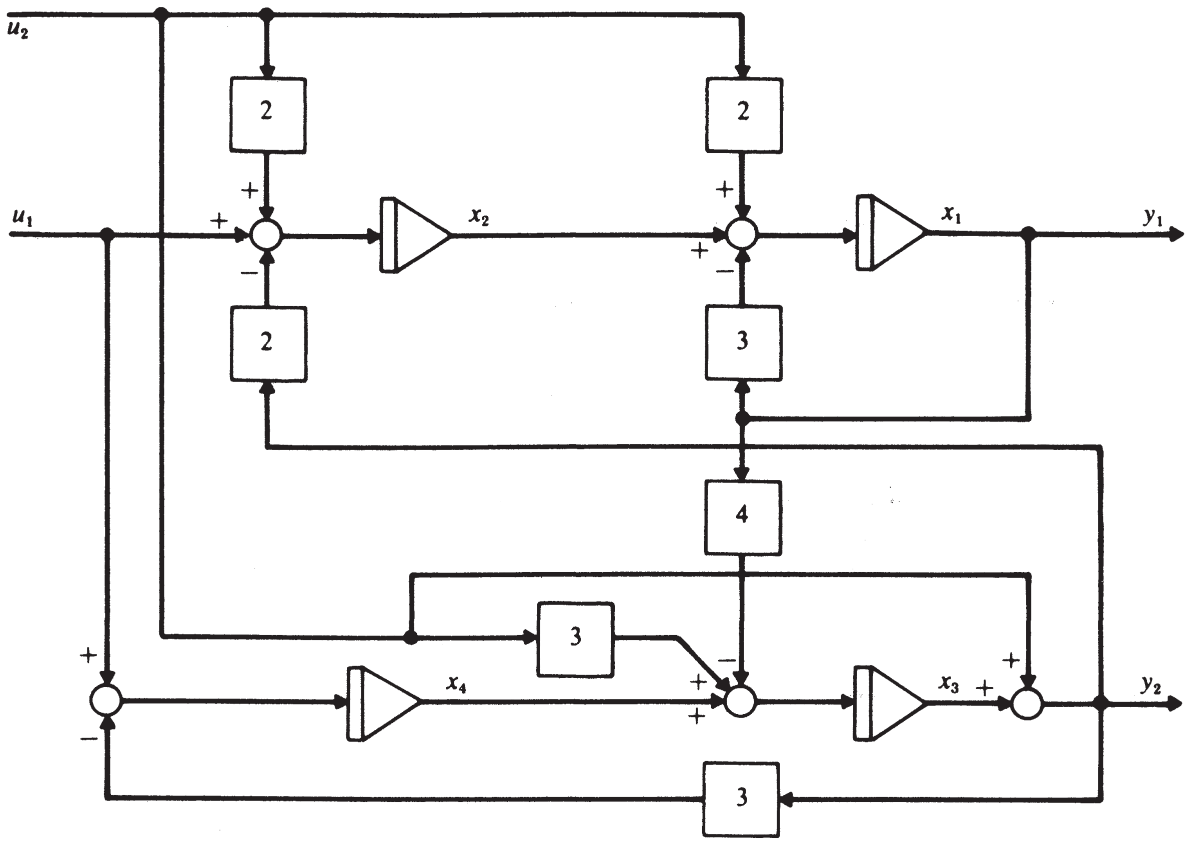


Figure 3.30

3.3 A single-input, single-out system has the transfer function

$$\frac{y(s)}{u(s)} = \frac{1}{s^3 + 10s^2 + 27s + 18} = T(s)$$

Find three different state variable representations.

(a) The transfer function represents the differential equation $\dddot{y} + 10\ddot{y} + 27\dot{y} + 18y = u$. Setting $x_1 = y$, $x_2 = \dot{y}$, $x_3 = \ddot{y}$ gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -18 & -27 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \quad \text{and} \quad y = [1 \quad 0 \quad 0]\mathbf{x}$$

(b) In factored form $T(s) = 1/[(s + 6)(s + 1)(s + 3)]$, and a simulation diagram is given in Figure 3.31.

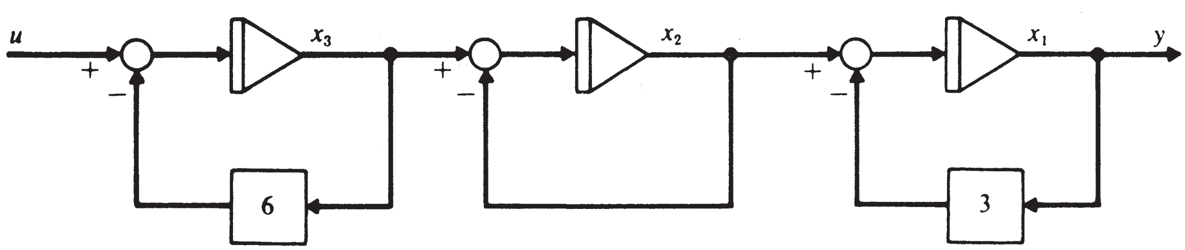


Figure 3.31

Then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -3 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \quad \text{and} \quad y = [1 \quad 0 \quad 0]\mathbf{x}$$

(c) Using partial fractions, $T(s) = a/(s+6) + b/(s+1) + c/(s+3)$, where

$$a = (s+6)T(s)|_{s=-6} = \tfrac{1}{15} \qquad c = (s+3)T(s)|_{s=-3} = -\tfrac{1}{6}$$

$$b = (s+1)T(s)|_{s=-1} = \tfrac{1}{10}$$

so the simulation diagram of Figure 3.32 is obtained.

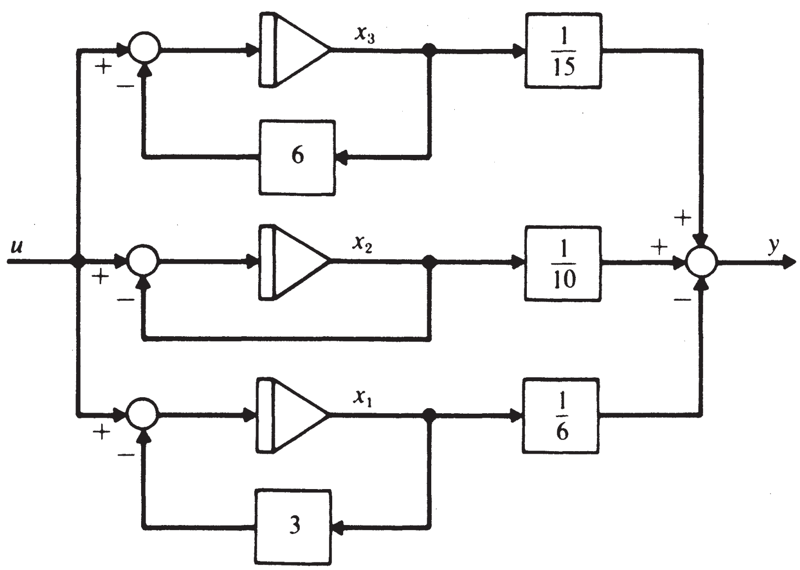


Figure 3.32

The state equations are

$$\dot{\mathbf{x}} = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u$$

$$y = [-\tfrac{1}{6} \quad \tfrac{1}{10} \quad \tfrac{1}{15}]\mathbf{x}$$

Note that since Figure 3.33*a* and *b* are equivalent, an alternative form for the matrices **B** and **C** is $\mathbf{B} = [-\tfrac{1}{6} \quad \tfrac{1}{10} \quad \tfrac{1}{15}]^T$ and $\mathbf{C} = [1 \quad 1 \quad 1]$.

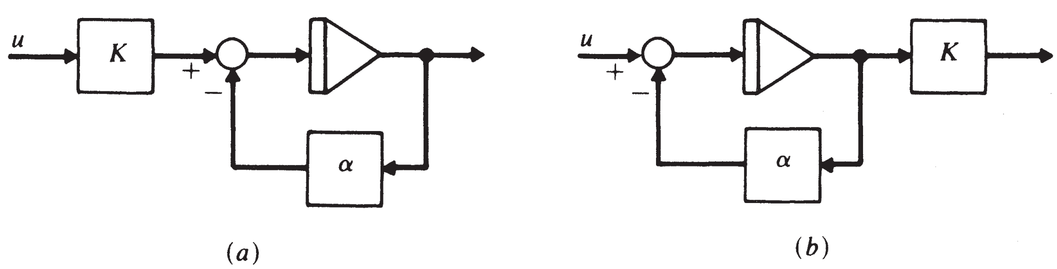


Figure 3.33

3.4 A system input-output transfer function is $T(s) = 1/[s^2(s+3)^3(s+1)]$. Find a state variable representation, using the partial fraction expansion of $T(s)$.

The expansion is

$$T(s) = a_1/s^2 + a_2/s + a_3/(s+3)^3 + a_4/(s+3)^2 + a_5/(s+3) + a_6/(s+1)$$

where

$$a_1 = s^2 T(s)|_{s=0} = \frac{1}{27} \qquad a_4 = \frac{d}{ds}\{(s+3)^3 T(s)\}|_{s=-3} = -\frac{7}{108}$$

$$a_2 = \frac{d}{ds}\{s^2 T(s)\}|_{s=0} = -\frac{2}{27} \qquad a_5 = \frac{1}{2!}\frac{d^2}{ds^2}\{(s+3)^3 T(s)\}|_{s=-3} = -\frac{11}{216}$$

$$a_3 = (s+3)^3 T(s)|_{s=-3} = -\frac{1}{18} \qquad a_6 = (s+1)T(s)|_{s=-1} = \frac{1}{8}$$

Using this expansion gives the simulation diagram of Figure 3.34.

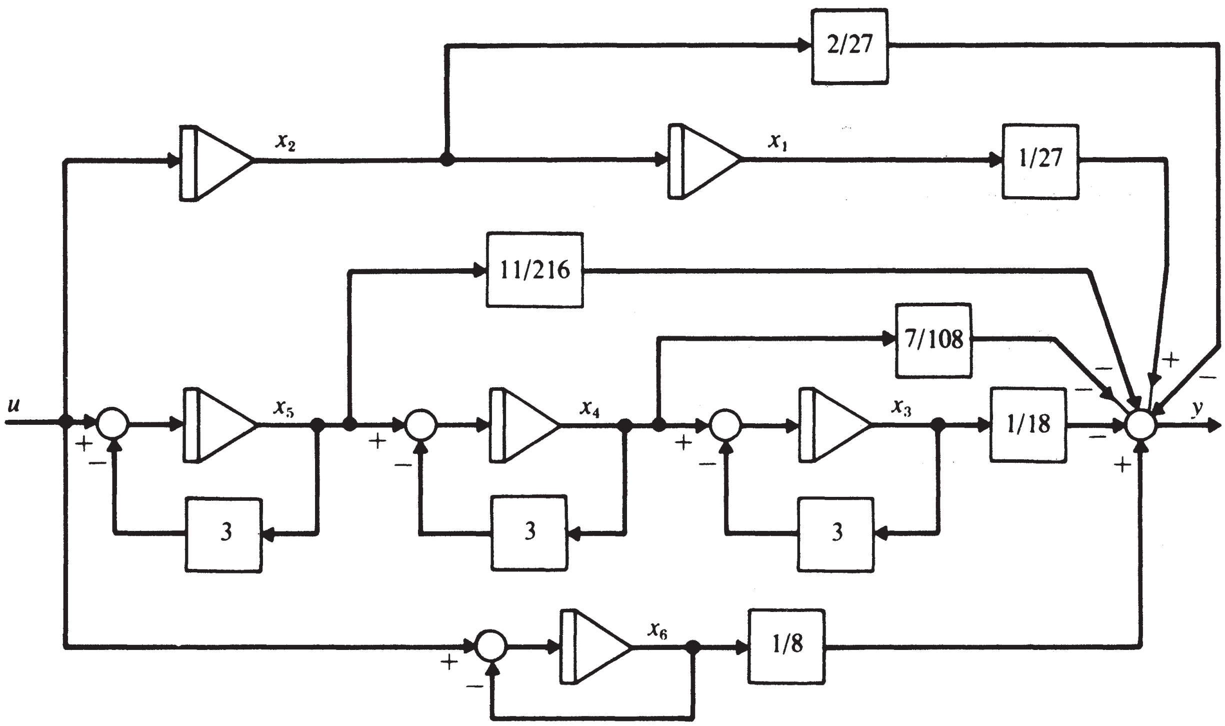


Figure 3.34

From the diagram,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 1 & 0 & 0 \\ 0 & 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}u \qquad \text{(note } \mathbf{A} \text{ is in Jordan form)}$$

$$y = \begin{bmatrix} \frac{1}{27} & -\frac{2}{27} & -\frac{1}{18} & -\frac{7}{108} & -\frac{11}{216} & \frac{1}{8} \end{bmatrix}\mathbf{x}$$

3.5 Find the state space representation for a system described by $T(s) = (s+1)/(s^2+7s+6)$.

Even though $T(s) = (s+1)/[(s+1)(s+6)]$, the common factor should not be canceled, or the system will be mistaken for a first-order system. Rather, use $\ddot{y} + 7\dot{y} + 6y = \dot{u} + u$, from which $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -7 & 1 \\ -6 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$ and $y = [1 \quad 0]\mathbf{x}$.

Linear Discrete-Time State Equations

3.6 A system has three inputs, $u_1(k)$, $u_2(k)$, and $u_3(k)$, and three outputs, $y_1(k)$, $y_2(k)$, and $y_3(k)$. The input-output difference equations are

$$y_1(k+3) + 6[y_1(k+2) - y_3(k+2)] + 2y_1(k+1) + y_2(k+1) + y_1(k) - 2y_3(k) = u_1(k) + u_2(k+1) \quad (1)$$

$$y_2(k+2) + 3y_2(k+1) - y_1(k+1) + 5y_2(k) + y_3(k) = u_1(k) + u_2(k) + u_3(k) \quad (2)$$

$$y_3(k+1) + 2y_3(k) - y_2(k) = u_3(k) - u_2(k) + 7u_3(k+1) \quad (3)$$

Draw a simulation diagram, select state variables, and write the matrix state equations.

Delaying each term in equation (*1*) three times gives

$$y_1(k) = \mathcal{D}\{-6[y_1(k) - y_3(k)] + \mathcal{D}[u_2(k) - 2y_1(k) - y_2(k) + \mathcal{D}[u_1(k) - y_1(k) + 2y_3(k)]]\}$$

Delaying each term in equation (*2*) twice gives

$$y_2(k) = \mathcal{D}\{-3y_2(k) + y_1(k) + \mathcal{D}[u_1(k) + u_2(k) + u_3(k) - 5y_2(k) - y_3(k)]\}$$

and from equation (*3*), delayed once,

$$y_3(k) = 7u_3(k) + \mathcal{D}\{u_3(k) - u_2(k) - 2y_3(k) + y_2(k)\}$$

The simulation diagram can be represented as shown in Figure 3.35.

Labeling x_1 through x_6 as shown in Figure 3.35 gives

$$\mathbf{x}(k+1) = \begin{bmatrix} -6 & 1 & 0 & 0 & 0 & 6 \\ -2 & 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & 0 & -5 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -2 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0 & 0 & 42 \\ 0 & 1 & 0 \\ 1 & 0 & 14 \\ 0 & 0 & 0 \\ 1 & 1 & -6 \\ 0 & -1 & -13 \end{bmatrix}\begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

and

$$\mathbf{y}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 7 \end{bmatrix}\begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

3.7 A system is described by the input-output equation

$$y(k+3) + 2y(k+2) + 4y(k+1) + y(k) = u(k)$$

This case is analogous to the simplest continuous-time problem where the input is not differentiated. Consequently, state variables can be selected as the output $y(k)$ and the output advanced by one and by two time steps. That is,

$$x_1(k) = y(k), \qquad x_2(k) = y(k+1), \qquad x_3(k) = y(k+2)$$

Then

$$x_1(k+1) = x_2(k), \qquad x_2(k+1) = x_3(k)$$

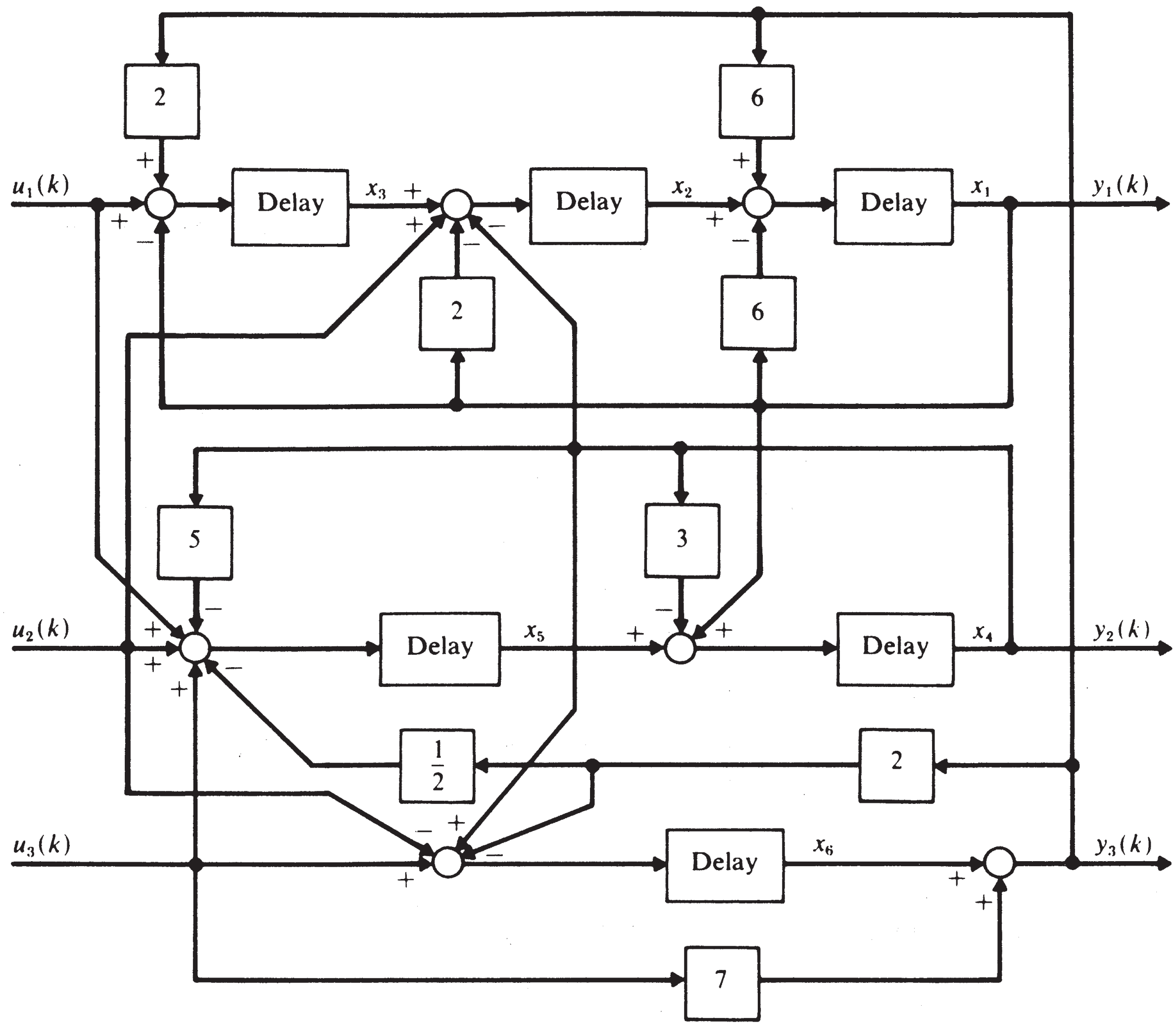


Figure 3.35

The final component of the state vector equation comes from the original difference equation and is

$$x_3(k+1) = -2x_3(k) - 4x_2(k) - x_1(k) + u(k)$$

Although its use is unnecessary in this simple problem, a possible simulation diagram is shown in Figure 3.36.

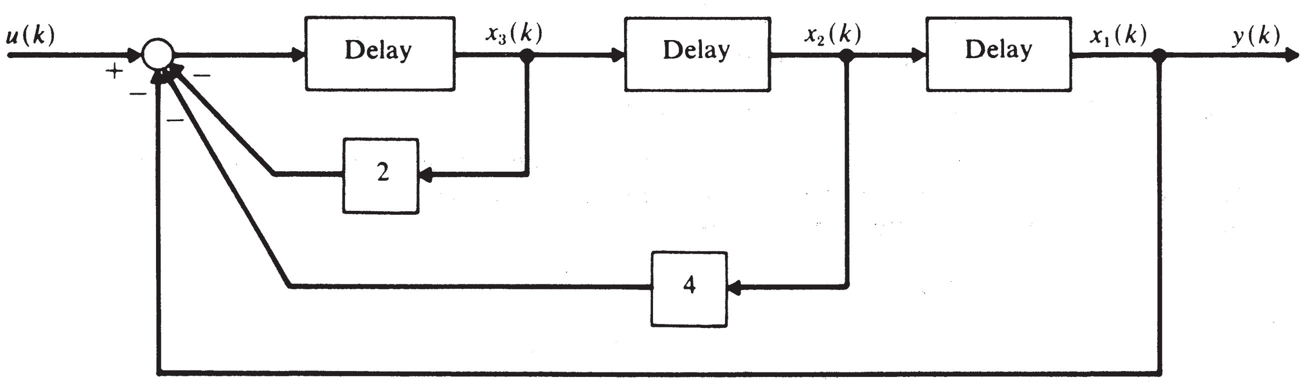


Figure 3.36

The state equations for this example are

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -4 & -2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0 \quad 0]\mathbf{x}(k) + 0u(k)$$

Time-Varying Coefficients

3.8 Obtain a state variable representation for the linear system with time-varying coefficients $\ddot{y} + e^{-t^2}\dot{y} + e^t y = u$.

Let $x_1 = y$, $x_2 = \dot{y}$, then $\dot{x}_2 = \ddot{y}$, so

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -e^t & -e^{-t^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad \text{and} \quad y = [1 \quad 0]\mathbf{x}$$

3.9 Apply the integration method of Section 3.4.3 to find a state variable model for

$$\dot{y} + a(t)y = b_0(t)u + b_1 \dot{u}$$

Solving for $\dot{y}$ and integrating once gives

$$y(t) = \int b_1 \dot{u}\, dt + \int [b_0 u - ay]\, dt$$

Using formal integration by parts on the first integral gives $\int b_1 \dot{u}\, dt = b_1 u(t) - \int \dot{b}_1 u\, dt$, so that $y(t) = b_1 u + \int [b_0 u - ay - \dot{b}_1 u]\, dt$. The simulation diagram of Figure 3.37 is drawn from this.

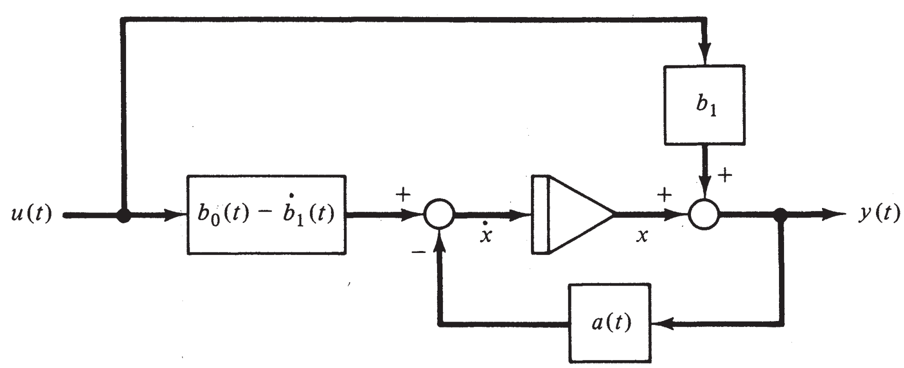


Figure 3.37

Using the integrator output as $x = y - b_1 u$ gives the state equations

$$\dot{x}(t) = -a(t)x(t) + [b_0(t) - \dot{b}_1(t) - b_1(t)a(t)]u(t)$$

$$y(t) = x(t) + b_1(t)u(t)$$

3.10 Find a state variable model for the second-order time-varying system

$$\ddot{y} + a_1(t)\dot{y} + a_0(t)y = b_0(t)u + b_1(t)\dot{u} + b_2(t)\ddot{u}$$

Solving for $\ddot{y}$ and integrating twice gives

$$y(t) = \iint \{b_0 u - a_0 y\}\, dt\, dt' + \int \left\{ \int [b_1 \dot{u} - a_1 \dot{y}]\, dt \right\} dt' + \iint b_2 \ddot{u}\, dt\, dt'$$

The first integrand on the right contains no derivatives and therefore is in final form. The second term on the right contains first derivatives of y and u, so integration by parts is needed to eliminate them. The innermost integral becomes

$$\int [b_1 \dot{u} - a_1 \dot{y}]\,dt = b_1 u - a_1 y - \int [\dot{b}_1 u - \dot{a}_1 y]\,dt$$

Since the third term in the expression for $y(t)$ contains $\ddot{u}$, integration by parts must be used twice.

$$\iint b_2 \ddot{u}\,dt\,dt' = \int \left\{ b_2 \dot{u} - \int \dot{b}_2 \dot{u}\,dt \right\} dt'$$

$$= b_2 u - \int \dot{b}_2 u\,dt - \int \left[\dot{b}_2 u - \int \ddot{b}_2 u\,dt \right] dt'$$

Recombining all terms into one nested integrator equation gives

$$y(t) = b_2 u + \int \left\{ b_1 u - a_1 y - 2\dot{b}_2 u + \int [b_0 u - a_0 y - \dot{b}_1 u + \dot{a}_1 y + \ddot{b}_2 u]\,dt \right\} dt'$$

The simulation diagram in Figure 3.38 is drawn from this. The state equations are

$$\dot{\mathbf{x}} = \begin{bmatrix} -a_1(t) & 1 \\ -[a_0(t) - \dot{a}_1(t)] & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} b_1 - 2\dot{b}_2 - a_1 b_2 \\ b_0 - \dot{b}_1 + \ddot{b}_2 - b_2(a_0 - \dot{a}_1) \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0]\mathbf{x}(t) + b_2(t)u(t)$$

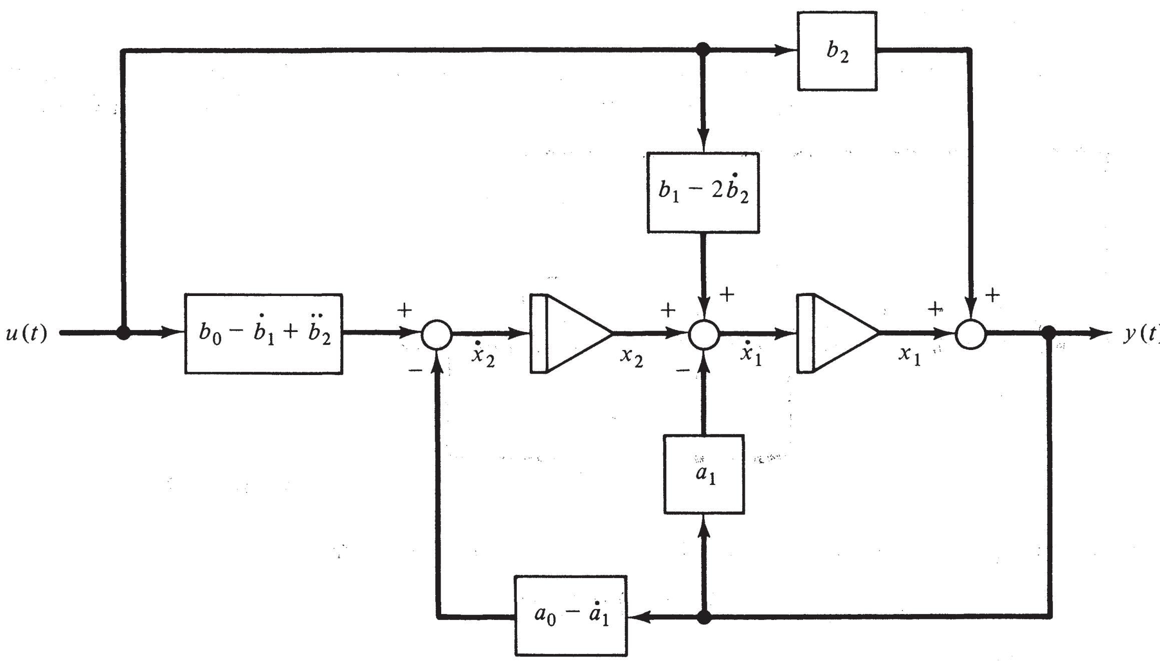


Figure 3.38

Linear Graph Method

3.11 Write the differential equations for the circuit of Figure 3.39 in state variable form. Consider the voltage across R_3 as the output.

The tree selected is shown in Figure 3.40.

The state variables are selected as the capacitor voltage x_1 and the inductor current x_2, so that $\dot{x}_1 = (1/C)i_C$, $\dot{x}_2 = (1/L)(v_1 - v_2)$. To express i_C, v_1, and v_2 in terms of x_1, x_2 and inputs, use

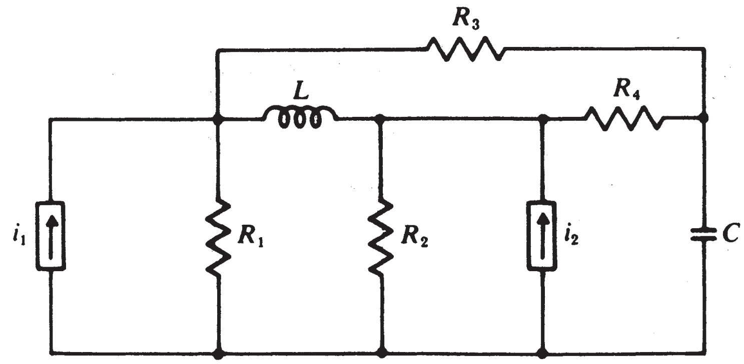


Figure 3.39

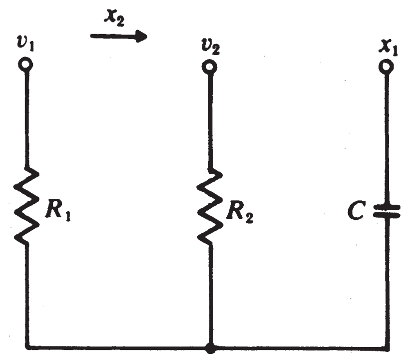


Figure 3.40

$$
\begin{aligned}
i_C &= i_{R_4} + i_{R_3} && \text{(cutset equation for tree branch } C\text{)}\\
i_{R_1} &= i_1 - i_{R_3} - x_2 && \text{(cutset equation for tree branch } R_1\text{)}\\
i_{R_2} &= x_2 + i_2 - i_{R_4} && \text{(cutset equation for tree branch } R_2\text{)}\\
R_3 i_{R_3} &= v_1 - x_1 && \text{(fundamental loop equation using link } R_3\text{)}\\
R_4 i_{R_4} &= v_2 - x_1 && \text{(fundamental loop equation using link } R_4\text{)}\\
v_1 &= R_1 i_{R_1}\\
v_2 &= R_2 i_{R_2}
\end{aligned}
$$

These are seven equations with seven unknowns. Solving for all the resistor currents gives

$$
i_{R_1} = \frac{R_3}{R_1 + R_3} i_1 + \frac{x_1}{R_1 + R_3} - \frac{R_3 x_2}{R_1 + R_3} \qquad i_{R_3} = -\frac{x_1}{R_1 + R_3} - \frac{R_1 x_2}{R_1 + R_3} + \frac{R_1 i_1}{R_1 + R_3}
$$

$$
i_{R_2} = \frac{x_1}{R_2 + R_4} + \frac{R_4 x_2}{R_2 + R_4} + \frac{R_4 i_2}{R_2 + R_4} \qquad i_{R_4} = -\frac{x_1}{R_2 + R_4} + \frac{R_2 x_2}{R_2 + R_4} + \frac{R_2 i_2}{R_2 + R_4}
$$

Using these to determine i_C, v_1, and v_2 gives

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-(R_1 + R_2 + R_3 + R_4)}{C(R_1 + R_3)(R_2 + R_4)} & \dfrac{R_2 R_3 - R_1 R_4}{C(R_2 + R_4)(R_1 + R_3)} \\ \dfrac{(R_1 R_4 - R_2 R_3)}{L(R_1 + R_3)(R_2 + R_4)} & -\dfrac{1}{L}\left[\dfrac{R_1 R_3}{R_1 + R_3} + \dfrac{R_2 R_4}{R_2 + R_4}\right] \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
$$

$$
+ \begin{bmatrix} \dfrac{R_1}{C(R_1 + R_3)} & \dfrac{R_2}{C(R_2 + R_4)} \\ \dfrac{R_1 R_3}{L(R_1 + R_3)} & \dfrac{-R_2 R_4}{L(R_2 + R_4)} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
$$

where $u_1 = i_1$, $u_2 = i_2$ are the inputs. The output is

$$
y = R_3 i_{R_3} = \begin{bmatrix} \dfrac{-R_3}{R_1 + R_3} & \dfrac{-R_1 R_3}{R_1 + R_2} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \dfrac{R_1 R_3}{R_1 + R_3} & 0 \end{bmatrix} \mathbf{u}
$$

3.12 Describe the hydraulic system of Problem 1.4, page 19, in state variable form.

The linear graph is redrawn as Fig. 3.41 using the analogous *RLC* symbols for the elements.

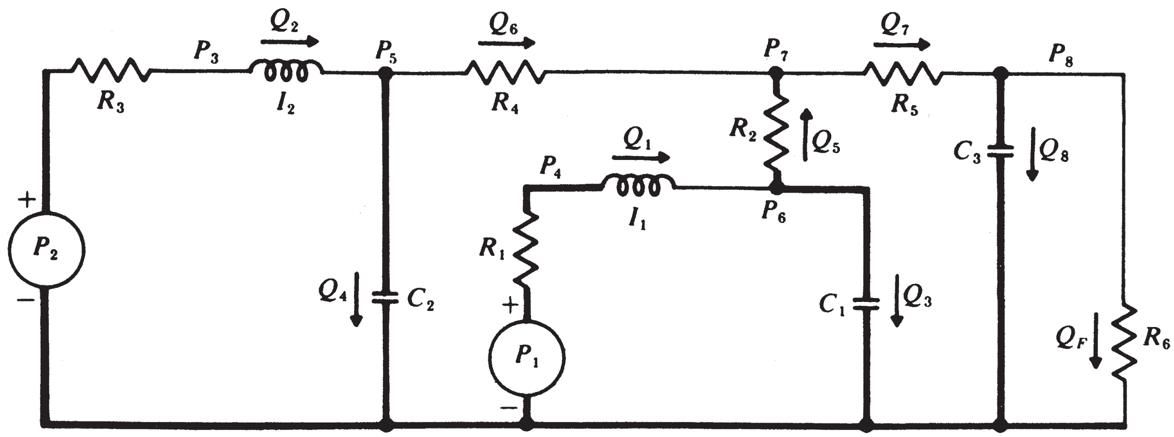


Figure 3.41

The tree is shown in heavy lines. The state variables are chosen as the pressures $x_1 = P_6$, $x_2 = P_5$, $x_3 = P_8$ and the flow rates $x_4 = Q_2$, $x_5 = Q_1$. Then $\dot{x}_1 = (1/C_1)\,Q_3$, $\dot{x}_2 = (1/C_2)\,Q_4$, $\dot{x}_3 = (1/C_3)\,Q_8$, $\dot{x}_4 = (1/I_2)\,P_{35}$, $\dot{x}_5 = (1/I_1)\,P_{46}$.

The last two are the easiest to complete and this is done first: $P_{35} = P_2 - R_3x_4 - x_2$, $P_{46} = P_1 - R_1x_5 - x_1$.

In order to express Q_3, Q_4, and Q_8 in terms of the state variables, the following simultaneous equations must be solved:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & R_4 & R_5 \\ 0 & 0 & 0 & -R_2 & R_4 & 0 \end{array}\right] \left[\begin{array}{c} Q_3 \\ Q_4 \\ Q_8 \\ \hline Q_5 \\ Q_6 \\ Q_7 \end{array}\right] = \left[\begin{array}{c} x_5 \\ x_4 \\ -x_3/R_6 \\ \hline 0 \\ x_2 - x_3 \\ x_2 - x_1 \end{array}\right]$$

The required solutions are

$$Q_3 = x_5 + \frac{1}{\Delta}[-R_4(x_2 - x_3) + (R_4 + R_5)(x_2 - x_1)]$$

$$Q_4 = x_4 + \frac{1}{\Delta}[-R_2(x_2 - x_3) - R_5(x_2 - x_1)]$$

$$Q_8 = -\frac{x_3}{R_6} + \frac{1}{\Delta}[(R_4 + R_2)(x_2 - x_3) - R_4(x_2 - x_1)]$$

where $\Delta = R_2R_4 + R_4R_5 + R_2R_5$. Using these relations allows the system equations to be put in the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, where $\mathbf{u} = [P_1 \quad P_2]^T$. The output is $Q_F = y$ and is given by $y = [0 \quad 0 \quad 1/R_6 \quad 0 \quad 0]\mathbf{x}$.

Nonlinear State Models

3.13 A schematic of a motor-generator system driving an inertia load J, with viscous damping b, at an angular velocity Ω is shown in Fig. 3.42. Derive a state space model of this system.

The pertinent equations are

1. $L_f\dfrac{di_f}{dt} + R_fi_f = e_f$

2. $e_g = f(i_f)$

3. $e_g - e_m = (R_g + R_m)i_m + (L_g + L_m)\dfrac{di_m}{dt}$

4. $T = K_mi_m$

5. $T = J\dot{\Omega} + b\Omega$

6. $e_m = K_m\Omega$

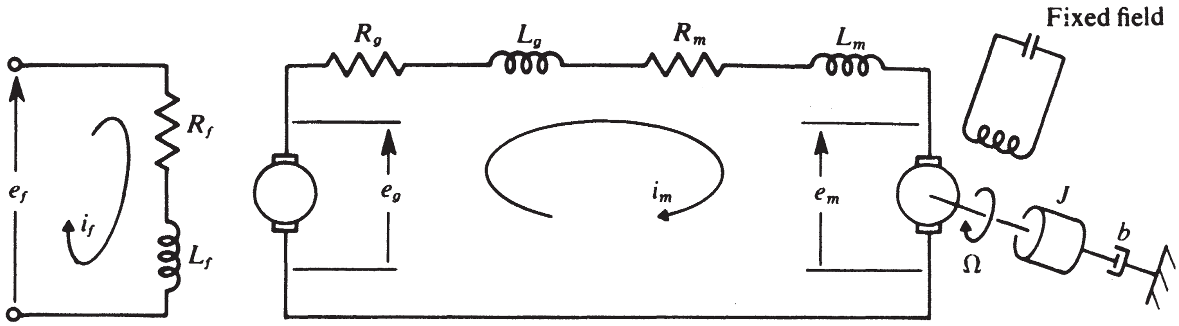


Figure 3.42

The simulation diagram of Fig. 3.43 can be constructed directly from these equations.

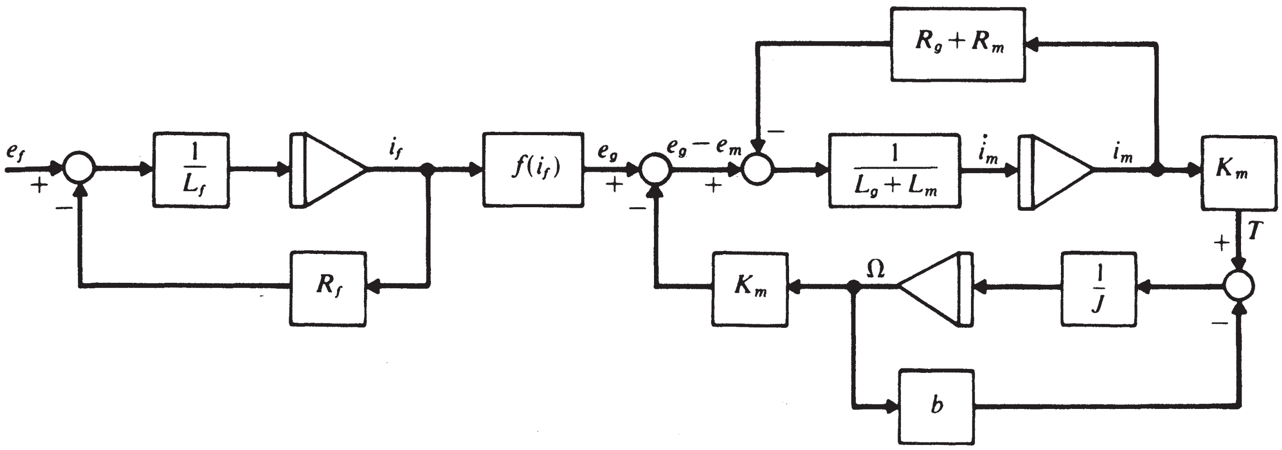


Figure 3.43

Selecting integrator outputs as state variables, $x_1 = \Omega$ (indicates kinetic energy in load), $x_2 = i_m$ (indicates magnetic energy in motor inductance), and $x_3 = i_f$ (indicates magnetic energy in the generator field), and letting the input be $e_f = u(t)$ gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{b}{J}x_1 + \frac{K_m}{J}x_2 \\ -\frac{K_m}{L_g + L_m}x_1 - \frac{R_g + R_m}{L_g + L_m}x_2 + \frac{f(x_3)}{L_g + L_m} \\ -\frac{R_f}{L_f}x_3 + \frac{u}{L_f} \end{bmatrix}$$

If it can be assumed that the generator characteristics are linear, i.e., if $e_g = K_g i_f$, then the above equations can be written in the standard linear form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$. If the speed Ω is considered to be the output y, then $y = [1 \quad 0 \quad 0]\mathbf{x}$.

3.14 Develop a state space model of a rocket vehicle (Figure 3.44) moving vertically above the earth. The vehicle thrust is $T = K\dot{m}$, where $\dot{m}$ is the rate of mass expulsion and can be controlled. Assume a drag force is given as a nonlinear function of velocity.

Letting the instantaneous mass of the vehicle be $m(t)$ and letting $D = f(\dot{h})$ be the drag, Newton's second law gives the dynamic force balance $m\ddot{h} = T(t) - f(\dot{h}) - [m(t)k^2 g_0/(k + h)^2]$, where an inverse square gravity law has been assumed. A simulation diagram is given in Figure 3.45.

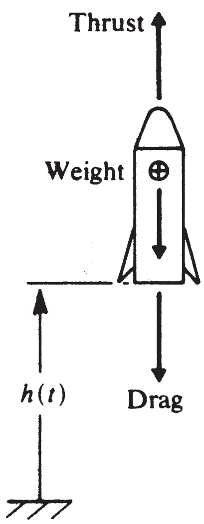


Figure 3.44

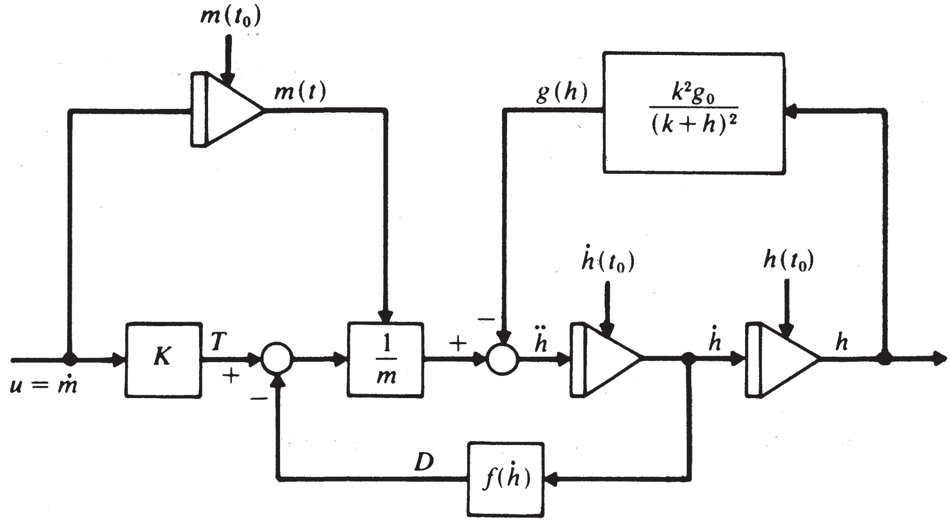


Figure 3.45

This is a nonlinear, time-variable system, but as before, outputs of the integrators are selected as state variables, $x_1 = h$, $x_2 = \dot{h}$, $x_3 = m$. Then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{Ku - f(x_2)}{x_3} - g(x_1) \\ u \end{bmatrix}$$

In this case it is not possible to obtain the linear form $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, but the preceding result is of the more general form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$.

3.15 Derive the equations of motion for a satellite rotating in free space under the influence of gas jets mounted along three mutually orthogonal body-fixed axes.

Let $\omega = [\omega_x \quad \omega_y \quad \omega_z]^T$ be the three components of angular velocity expressed with respect to the body-fixed axes. Let $\mathbf{T} = [T_x \quad T_y \quad T_z]^T$ be the three components of input control torques. Newton's second law, as applied to a rotating body, states that $d\mathbf{H}/dt = \mathbf{T}$, where $\mathbf{H}$ is the angular momentum vector, and the time rate of change d/dt is with respect to a fixed inertial reference. The vector $\mathbf{H}$ can be expressed in body coordinates as $\mathbf{H} = [J_x\omega_x \quad J_y\omega_y \quad J_z\omega_z]^T$, where the constants J_i are moments of inertia of the body and x, y, z are assumed to be principal axes of inertia. The inertial rate of change $d\mathbf{H}/dt$ is related to the apparent rate $[\dot{\mathbf{H}}]$ as seen by an observer moving with the body by $d\mathbf{H}/dt = [\dot{\mathbf{H}}] + \boldsymbol{\omega} \times \mathbf{H}$. Therefore,

$$T_x = J_x\dot{\omega}_x + (J_z - J_y)\omega_y\omega_z, \qquad T_y = J_y\dot{\omega}_y + (J_x - J_z)\omega_x\omega_z,$$

$$T_z = J_z\dot{\omega}_z + (J_y - J_x)\omega_x\omega_y$$

These equations are often referred to as Euler's dynamical equations. Rearranging gives

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \dfrac{J_y - J_z}{J_x}\omega_y\omega_z \\ \dfrac{J_z - J_x}{J_y}\omega_x\omega_z \\ \dfrac{J_x - J_y}{J_z}\omega_x\omega_y \end{bmatrix} + \begin{bmatrix} \dfrac{T_x}{J_x} \\ \dfrac{T_y}{J_y} \\ \dfrac{T_z}{J_z} \end{bmatrix}$$

This obviously is in the state variable form and is linear in the control variables T_i but nonlinear in the state variables ω_i due to the products $\omega_i\,\omega_j$.

3.16 Apply the results of the preceding problem to the satellite of Figure 3.46, which is spin stabilized about the x axis. That is, $\omega_x = S$ is a large value, ω_y and $\omega_z << S$ represent small wobbling errors. Assume the satellite is rotationally symmetric about the x axis. Find a linear approximation for the state equation.

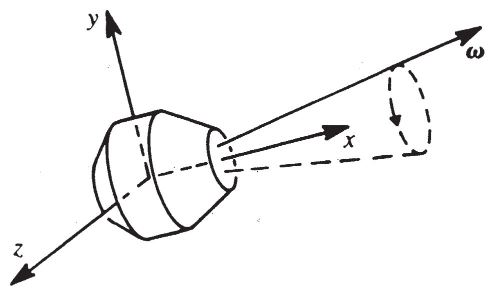


Figure 3.46

Due to symmetry $J_y = J_z$, so $\dot{\omega}_x = T_x/J_x$. If $T_x = 0$, in the absence of any other torques, $\omega_x =$ constant. If the definition $[(J_x - J_y)/J_z]S \triangleq \Omega$ is introduced, then the linear relations between the control torques T_y, T_z and the small wobbling errors ω_y, ω_z are

$$\begin{bmatrix} \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} 0 & -\Omega \\ \Omega & 0 \end{bmatrix} \begin{bmatrix} \omega_y \\ \omega_z \end{bmatrix} + \frac{1}{J_y} \begin{bmatrix} T_y \\ T_z \end{bmatrix}$$

3.17 A nonlinear time-varying second-order system is described by

$$\ddot{y} + g(y, \dot{y}, u, t) + f(y, t)\dot{u} = 0$$

Assume that the function $f(\ )$ is sufficiently smooth so that the first derivatives with respect to each of its arguments exist and are well behaved. Derive a state variable model for this system.

Solving for $\ddot{y}$ and integrating twice gives

$$y(t) = \iint -g(y, \dot{y}, u, \tau)\, d\tau\, dt' - \iint f(y, \tau)\dot{u}\, d\tau\, dt'$$

$$= \iint -g(y, \dot{y}, u, \tau)\, d\tau\, dt' - \int f(y, t')u\, dt' + \iint [u\, df/dt']\, d\tau\, dt'$$

The chain-rule expanded form $df/dt = [\partial f/\partial y]\dot{y} + [\partial f/\partial t]$ is used in the preceding equation, which is used to draw the simulation diagram shown in Figure 3.47. The blocks in this diagram are general functional evaluation boxes, and the outputs depend upon all the signals shown as inputs in a way described by the equation inside the box. The state equations cannot be written as simple matrix products as in the linear case.

$$\dot{x}_1 = x_2 - f(x_1, t)u(t)$$

$$\dot{x}_2 = u(t)[\partial f/\partial t + x_2\, \partial f/\partial x_1] - g(x_1, x_2, t, u)$$

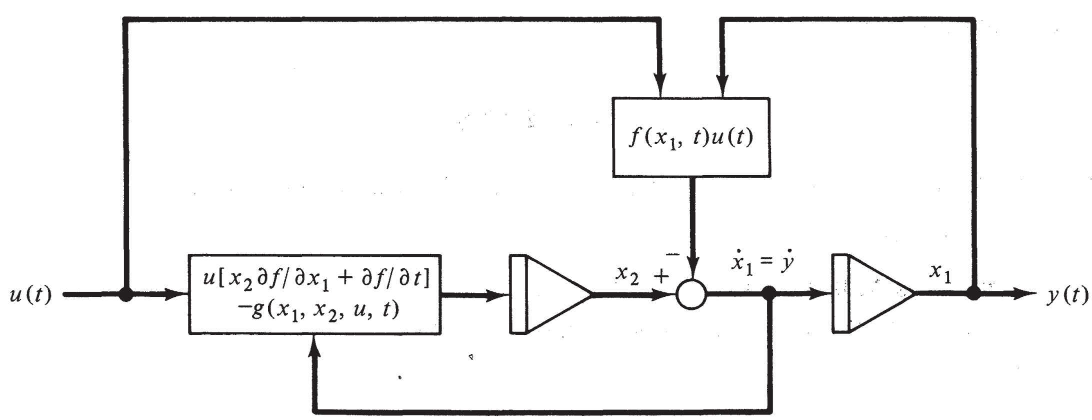


Figure 3.47

and

$$y = [1 \quad 0]\mathbf{x}$$

3.18 Draw a simulation diagram and express the following nonlinear, time-varying system in state space form:

$$y(k+3) + y(k+2)y(k+1) + \alpha \sin(\omega k) y^2(k) + y(k) = u(k) - 3u(k+1)$$

Rewriting the equation as

$$y(k+3) + 3u(k+1) = u(k) - y(k+2)y(k+1) - \alpha \sin(\omega k) y^2(k) - y(k)$$

allows the simulation diagram to be drawn as shown in Figure 3.48.

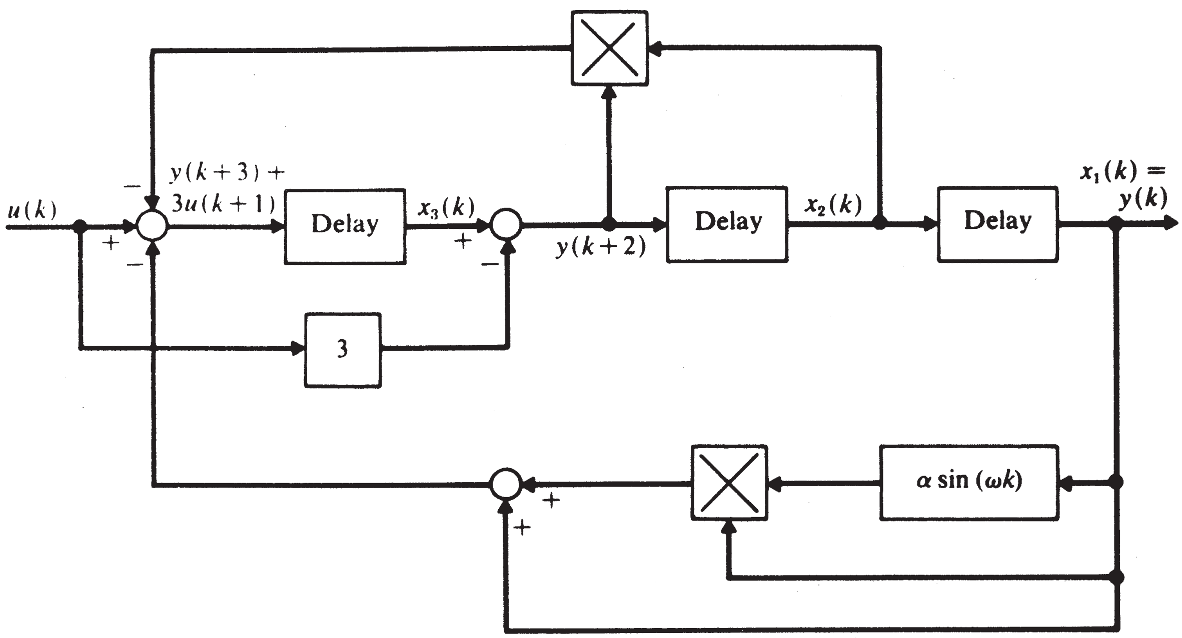


Figure 3.48

Labeling x_1 through x_3 as shown, the state equations are

$$x_1(k+1) = x_2(k)$$

$$x_2(k+1) = x_3(k) - 3u(k)$$

$$x_3(k+1) = -x_2(k)x_3(k) + 3x_2(k)u(k) - \alpha \sin(\omega k) x_1^2(k) - x_1(k) + u(k)$$

and the output equation is $y(k) = [1 \quad 0 \quad 0]\mathbf{x}(k)$.

PROBLEMS

3.19 Convince yourself that both Figure 3.49*a* and *b* represent the system described by

$$\ddot{y} + a\dot{y} + by = u$$

and find the matrices **A, B, C,** and **D** for each case.

3.20 Find a state space representation for the system described by

$$\dot{y}_1 + 3(y_1 + y_2) = u_1$$

$$\ddot{y}_2 + 4\dot{y}_2 + 3y_2 = u_2$$

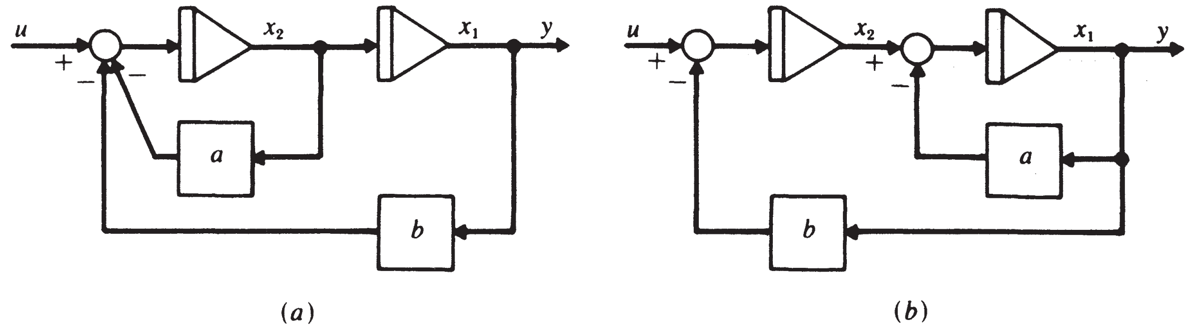


Figure 3.49

3.21 Find a state space representation for the system described by

$$\ddot{y}_1 + 3\dot{y}_1 + 2(y_1 - y_2) = u_1 + \dot{u}_2$$

$$\dot{y}_2 + 3(y_2 - y_1) = u_2 + 2\dot{u}_1$$

3.22 Find the state variable equations for a system described by

$$T(s) = 1/(s^3 + 8s^2 + 13s + 6)$$

using the partial fraction expansion.

3.23 Describe the circuit of Figure 3.50 in state variable form.

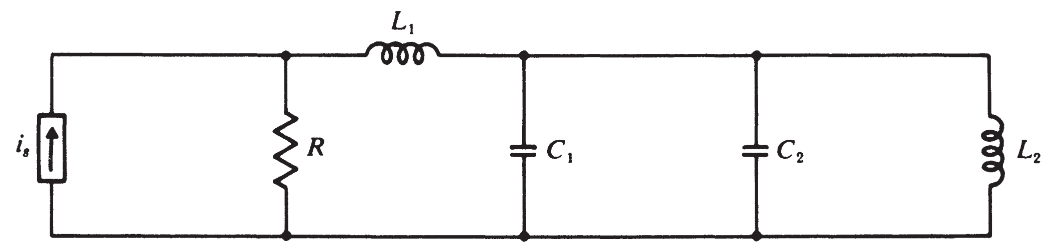


Figure 3.50

3.24 Represent the circuit of Figure 3.51 in state variable form. Assume that the amplifier is an ideal voltage amplifier, that is $v_5 = Kv_4$ and the amplifier draws no current. The transformer ratio is N. The output is the voltage across C_2.

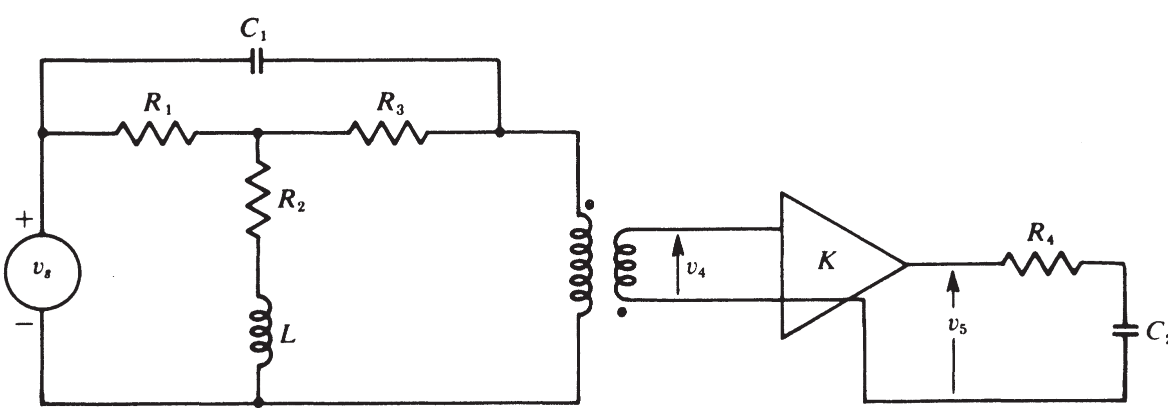


Figure 3.51

3.25 Use as state variables the voltages x_1 and x_2 and the current x_3 as shown in Figure 3.52. Derive the state equations, letting $v_s = u_1$ and $i_s = u_2$.

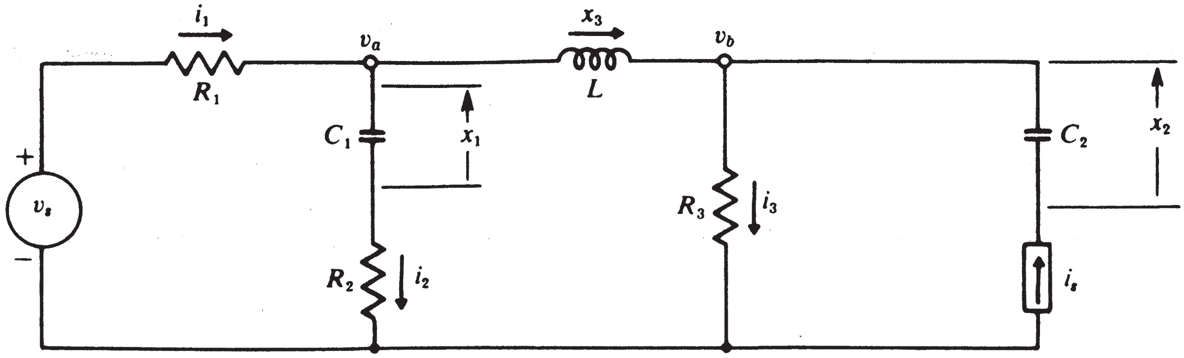


Figure 3.52

3.26 A system has two inputs and two outputs. The input-output equations are

$$y_1(k+2) + 10y_1(k+1) - y_2(k+1) + 3y_1(k) + 2y_2(k) = u_1(k) + 2u_1(k+1)$$

$$y_2(k+1) + 4[y_2(k) - y_1(k)] = 2u_2(k) - u_1(k)$$

Select state variables and write the vector matrix state equations.

3.27 Draw two different simulation diagrams and obtain two different state variable representations for the system described by

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