CHAPTER

State-Space Modeling

8

OBJECTIVES

This chapter focuses on the state-space approach to modeling dynamic systems in the time domain and determining the corresponding solution. The subject matter of the chapter is related to the modeling of Chapters 2—5, as well as with the Laplace transform of Chapter 6 and the transfer function of Chapter 7. The following topics are studied:

- The state-space concept as a matrix procedure for rendering the time-domain dynamic models of SISO (single-input, single-output) and MIMO (multiple-input, multiple-output) systems into first-order differential equations and for obtaining solutions to the corresponding models.
- Deriving state-space models directly in the time domain or via transfer functions in the Laplace domain when the input includes time derivatives in it.
- Analytical and MATLAB methods of converting between transfer function or zero-pole-gain models and state-space models.
- Nonlinear state-space models and procedures for linearizing such models.
- The use of state-space models to solve for the free response with nonzero initial conditions and for the forced response of dynamic systems.
- Specialized MATLAB commands designed to model and solve state-space problems.
- Application of Simulink to graphically model and solve state-space system dynamics problems.

INTRODUCTION

This chapter introduces the state-space modeling method for SISO and MIMO dynamic systems into the time domain. The approach is a matrix method of converting large-order differential equations into an equivalent number of first-order differential equations. The state-space methodology is able to model systems with a large number of degrees of freedom, as well as systems with nonlinearities. Similar to the transfer function approach, which is the subject of Chapter 7, the material presented here focuses on deriving state-space models of dynamic systems and solving these models to determine the time response using analytical methods, MATLAB custom commands, or Simulink. While the transfer function model belongs to the

System Dynamics for Engineering Students. http://dx.doi.org/10.1016/B978-0-12-804559-6.00008-7

Laplace domain, an state-space model operates in the time domain. The state-space approach utilizes the same matrix model for both SISO and MIMO dynamic systems. Conversions between state-space and transfer function or zero-pole-gain models are also studied here.

8.1 THE CONCEPT AND MODEL OF THE STATE-SPACE APPROACH

The state-space approach employs a unique matrix formulation to model a dynamic system and to evaluate the forced response of SISO and MIMO systems. This model is derived from the particular mathematical method, which converts generally higher-order differential equations into an equivalent number of first-order differential equations, which are easier to solve. Consider, for instance, the following linear third-order differential equation with constant coefficients, the unknown $y(t)$, and the input (forcing) $u(t)$:

$$\begin{aligned} &a_3 \cdot \dddot{y}(t) + a_2 \cdot \ddot{y}(t) + a_1 \cdot \dot{y}(t) + a_0 \cdot y(t) = u(t) \\ &\text{or } \dddot{y}(t) = -\frac{a_0}{a_3} \cdot y(t) - \frac{a_1}{a_3} \cdot \dot{y}(t) - \frac{a_2}{a_3} \cdot \ddot{y}(t) + \frac{1}{a_3} \cdot u(t) \end{aligned} \tag{8.1}$$

This equation can be converted into three first-order differential equations by using the following new variables:

$$x_1(t) = y(t); \;\; x_2(t) = \dot{y}(t); \;\; x_3(t) = \ddot{y}(t) \tag{8.2}$$

The following three equations are obtained from Eqs. (8.1) and (8.2):

$$\begin{aligned} &\dot{x}_1(t) = x_2(t) \\ &\dot{x}_2(t) = x_3(t) \\ &\dot{x}_3(t) = -\frac{a_0}{a_3} \cdot x_1(t) - \frac{a_1}{a_3} \cdot x_2(t) - \frac{a_2}{a_3} \cdot x_3(t) + \frac{1}{a_3} \cdot u(t) \end{aligned} \tag{8.3}$$

The three Eq. (8.3) can be written in matrix form as:

$$\begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0/a_3 & -a_1/a_3 & -a_2/a_3 \end{bmatrix} \cdot \begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 1/a_3 \end{Bmatrix} \cdot u(t) \tag{8.4}$$

Eq. (8.4) allows solving for the new time-dependent variables of Eq. (8.2). The unknown of the original differential Eq. (8.1) is simply found from the first Eq. (8.2) as $y(t) = x_1(t)$, but it can also be expressed in a form similar to Eq. (8.4) as:

$$y(t) = \{1 \quad 0 \quad 0\} \cdot \begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} + 0 \cdot u(t) \tag{8.5}$$

As we shall see shortly, Eq. (8.4) that allows solving for the new variables $x_1(t)$, $x_2(t)$, $x_3(t)$ in terms of the input $u(t)$ together with Eq. (8.5) that yields the original output $y(t)$ in terms of $x_1(t)$, $x_2(t)$, $x_3(t)$, and $u(t)$ form the *state-space model* of the original mathematical model given in Eq. (8.1).

While the example previously analyzed looked at a SISO system, it is also possible to use the same conceptual model represented by Eqs. (8.4) and (8.5) to describe the behavior of MIMO systems with a large number of outputs (or degrees of freedom (DOF)). These systems result in an equally large number of differential equations that need to be solved for these unknown outputs. Eqs. (8.4) and (8.5) are generalized as:

$$\begin{aligned} \{\dot{x}(t)\} &= [A]\cdot\{x(t)\} + [B]\cdot\{u(t)\} \\ \{y(t)\} &= [C]\cdot\{x(t)\} + [D]\cdot\{u(t)\} \end{aligned} \tag{8.6}$$

The first Eq. (8.6), which is a generalization of Eq. (8.4), is known as the *state equation*, and by solving it, the *state vector* $\{x(t)\}$ is determined. The output vector $\{y(t)\}$ is subsequently calculated from the *output equation*, which is the second Eq. (8.6)—a generalization of the particular Eq. (8.5). The *standard state-space model* consists of these two equations. The four matrices in Eqs. (8.6) contain parameters of the dynamic system being studied; they are $[A]$, the *state matrix*, $[B]$, the *input matrix*, $[C]$, the *output matrix*, and $[D]$, the *direct transmission matrix*. Proper selection of the *state variables* (which are collected in *state vectors* forming an state-space) ensures that Eq. (8.6) can always be applied, irrespective of the number of DOF and of the system order. This topic is analyzed in the next section of this chapter.

As Eq. (8.6) indicates, the input and output vectors need not have the same dimensions. Assuming the input vector has the dimension m and the output vector has the dimension p, the dimensions of the other matrices can be determined based also on the dimension of the state vector, which is n. The first Eq. (8.6) shows that $[A]$ is square with n rows and n columns. Similarly, the matrix $[B]$ is of $n \times m$ dimension, as it results from the same Eq. (8.6). From the second Eq. (8.6), it follows that $[C]$ has $p \times n$ dimension, whereas the matrix $[D]$ is of $p \times m$ dimension. A visual representation of the four matrices' dimensions is given in Figure 8.1.

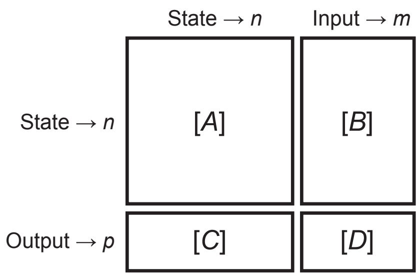

FIGURE 8.1

Dimensions of State-Space Matrices.

Nonuniqueness of a State-Space Model

Figure 8.1 visually suggests that for any given dynamic system (defined by m inputs and p outputs) one can use different values for n—the state variables—and still obtain a valid state-space model. The direct inference is that there are more than one state-space model for any given dynamic system model. Moreover, two state-space models of the same system may have the same number of state variables n as argued next, and the conclusion is that state-space models are not unique. Let us assume a vector $\{x(t)\}$ has been found to be a state vector, based on which a state-space model has been derived for a specific dynamic system, and consider the following transformation:

$$\{x(t)\} = [H] \cdot \{\bar{x}(t)\} \tag{8.7}$$

where $[H]$ is a nonsingular square matrix, and $\{\bar{x}(t)\}$ is another vector. Substitution of $\{x(t)\}$ of Eq. (8.7) into the standard state equation—first Eq. (8.6)—results in:

$$[H] \cdot \left\{\dot{\bar{x}}(t)\right\} = [A] \cdot [H] \cdot \{\bar{x}(t)\} + [B] \cdot \{u(t)\} \tag{8.8}$$

Left multiplying in Eq. (8.8) by $[H]^{-1}$ produces:

$$\left\{\dot{\bar{x}}(t)\right\} = [H]^{-1} \cdot [A] \cdot [H] \cdot \{\bar{x}(t)\} + [H]^{-1} \cdot [B] \cdot \{u(t)\} \tag{8.9}$$

Similar substitution of $\{x(t)\}$ from Eq. (8.7) into the generic output equation of the state-space model—the second Eq. (8.6)—yields:

$$\{y(t)\} = [C] \cdot [H] \cdot \{\bar{x}(t)\} + [D] \cdot \{u(t)\} \tag{8.10}$$

Eqs. (8.9) and (8.10) can be rewritten as follows:

$$\begin{aligned} \left\{\dot{\bar{x}}(t)\right\} &= \left[\bar{A}\right] \cdot \{\bar{x}(t)\} + \left[\bar{B}\right] \cdot \{u(t)\} \\ \{\bar{y}(t)\} &= \left[\bar{C}\right] \cdot \{\bar{x}(t)\} + [D] \cdot \{u(t)\} \end{aligned} \tag{8.11}$$

with: $\left[\bar{A}\right] = [H]^{-1} \cdot [A] \cdot [H]$; $\left[\bar{B}\right] = [H]^{-1} \cdot [B]$; $\left[\bar{C}\right] = [C] \cdot [H]$. Eq. (8.11) is of the type given in Eq. (8.6), which represents an state-space model. As a consequence, Eq. (8.11) also represents a state-space model, and since there are an infinite number of nonsingular square matrices $[H]$, there are also an infinite number of state vectors that can be obtained from the original state vector $\{x(t)\}$ according to Eq. (8.7). Therefore, an infinite number of state-space models can be derived from any specific state-space model.

Solution of the State-Space Equations

There are various solution methods for solving the state equation—the first Eq. (8.6), which is a set of first-order differential equations—including the eigenvalue method (for homogeneous equations), the direct and inverse Laplace transform method, and matrix methods (such as the matrix exponential or the state-transition matrix). Section 8.3 utilizes the state-transition matrix method in conjunction with Laplace transforms to solve for the state vector $\{x(t)\}$. Laplace transforming Eq. (8.6) for zero initial conditions results in

$$\begin{cases} s \cdot \{X(s)\} = [A] \cdot \{X(s)\} + [B] \cdot \{U(s)\} \\ \{Y(s)\} = [C] \cdot \{X(s)\} + [D] \cdot \{U(s)\} \end{cases} \tag{8.12}$$

where $\{X(s)\}$, $\{U(s)\}$, and $\{Y(s)\}$ are the Laplace transforms of $\{x(t)\}$, $\{u(t)\}$, and $\{y(t)\}$, respectively. The first Eq. (8.12) yields $\{X(s)\}$ as

$$\{X(s)\} = (s \cdot [I] - [A])^{-1} \cdot [B] \cdot \{U(s)\} \tag{8.13}$$

which, substituted into the second Eq. (8.12) gives

$$\{Y(s)\} = [G(s)] \cdot \{U(s)\} \text{ with } [G(s)] = [C] \cdot (s \cdot [I] - [A])^{-1} \cdot [B] + [D] \tag{8.14}$$

$[G(s)]$ is the *transfer function matrix* that is introduced in Chapter 7. As we shall see in this chapter, Eq. (8.14) also serves at converting a state-space model into a transfer function model. Once the Laplace-domain output vector is determined from Eq. (8.14), the time-domain output is simply $\{y(t)\} = \mathcal{L}^{-1}[\{Y(s)\}] = \mathcal{L}^{-1}[[G(s)] \cdot \{U(s)\}]$. Let us consider a couple of examples illustrating how to derive the state-space models of dynamic systems from mathematical models.

Example 8.1

Derive a state-space model for a single-mesh *RLC* series electrical circuit formed of a resistor *R*, an inductor *L*, a capacitor *C*, and a voltage source *v*. The input is the source voltage *v*, and the output is the charge *q*.

Solution

As shown in Chapter 4, the following differential equation is the mathematical model of the *RLC* electrical system:

$$L \cdot \ddot{q} + R \cdot \dot{q} + \frac{1}{C} \cdot q = v \text{ or } \ddot{q} = -\frac{1}{L \cdot C} \cdot q - \frac{R}{L} \cdot \dot{q} + \frac{1}{L} \cdot v \tag{8.15}$$

where *L* is the inductance, *R* is the resistance, *C* is the capacitance, *q* is the charge, and *v* is the source voltage. Let us select the following state variables to form the state vector $\{x\}$:

$$x_1 = q; \quad x_2 = \dot{q}; \quad \{x\} = \{x_1 \quad x_2\}^T \tag{8.16}$$

In Eqs. (8.15) and (8.16) the independent variable *t* (time) has been dropped to simplify notation, but all variables in this example (q, v, x_1, and x_2) are functions of time. The second form of Eq. (8.15) and the particular choice of the state variables of Eq. (8.16) indicate that

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{1}{L \cdot C} \cdot x_1 - \dfrac{R}{L} \cdot x_2 + \dfrac{1}{L} \cdot v \end{cases} \tag{8.17}$$

The source voltage is the input to the system and therefore can be denoted by *u*, so $v = u$. Eq. (8.17) is collected in vector-matrix form as

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{1}{L \cdot C} & -\dfrac{R}{L} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \dfrac{1}{L} \end{Bmatrix} \cdot u \tag{8.18}$$

Comparing Eq. (8.18) to the first Eq. (8.6) indicates that Eq. (8.18) is the state equation; the state and input matrices are

$$[A] = \begin{bmatrix} 0 & 1 \\ -\dfrac{1}{L \cdot C} & -\dfrac{R}{L} \end{bmatrix}; \quad [B] = \begin{Bmatrix} 0 \\ \dfrac{1}{L} \end{Bmatrix} \tag{8.19}$$

If the charge is the output, which can be denoted formally as $y = q$, then the first Eq. (8.16) becomes $y = x_1$ and can be written in vector form as

$$y = \{1 \quad 0\} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + 0 \cdot u \tag{8.20}$$

Comparison of this equation with the second Eq. (8.6) indicates that Eq. (8.20) represents the output equation, and its defining matrices are

$$[C] = \{1 \quad 0\}; \quad [D] = 0 \tag{8.21}$$

The state Eq. (8.18) and the output Eq. (8.20) form the state-space model of the *RLC* electrical circuit.

Note: In Chapter 4, the *potential electrical energy* stored by a capacitor was shown to be equal to ½(q^2/C), whereas the *kinetic electrical energy* related to an inductor was ½ $L\dot{q}^2$. Both energies describe a given system state, defined as a function of charge (through the potential energy) and of charge derivative, or current (through the kinetic energy). The particular choice of the state variables as the charge and the charge time derivative, Eq. (8.16), indicates the connection between parameters with physical significance (the energy terms) and state variables. This observation is also valid for other systems, for instance, mechanical ones, where the potential energy depends on displacement and the kinetic energy on velocity (the displacement's first derivative); for a second-order differential equation representing the mathematical model of a mechanical system, a common choice of state variables consists in displacement and velocity. ■

Example 8.2

Find another state-space model for the electrical circuit of Example 8.1 using the current i and the voltage across the capacitor v_C as both state variables and outputs.

Solution

The Kirchhoff's voltage law (KVL) can be written in the form $v_R + v_L + v_C = v$, where v_R, v_L, and v_C are the voltages across the resistor, inductor, and capacitor, respectively. To simplify notation, we will again not use the explicit time dependency notation in both physical and state-space model variables. By taking into account known current—voltage relationships, the voltage equation becomes

$$R \cdot i + L \cdot \frac{di}{dt} + v_C = v \quad \text{or} \quad \frac{di}{dt} = -\frac{R}{L} \cdot i - \frac{1}{L} \cdot v_C + \frac{1}{L} \cdot u \tag{8.22}$$

where it has been considered that v is the input to the system; that is, $v = u$. Another linear independent relationship between i and v_C, with v_C appearing as a derivative is

$$\frac{dv_C}{dt} = \frac{1}{C} \cdot i \tag{8.23}$$

If we now select the state variables to be i and v_C, namely:

$$\bar{x}_1 = i; \quad \bar{x}_2 = v_C \tag{8.24}$$

Eqs. (8.22) and (8.23) can be formulated in vector-matrix form as a state equation:

$$\begin{Bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{Bmatrix} = \begin{bmatrix} -\dfrac{R}{L} & -\dfrac{1}{L} \\ \dfrac{1}{C} & 0 \end{bmatrix} \cdot \begin{Bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{Bmatrix} + \begin{Bmatrix} \dfrac{1}{L} \\ 0 \end{Bmatrix} \cdot u \tag{8.25}$$

Considering the output vector is defined by the components

$$\bar{y}_1 = i; \;\; \bar{y}_2 = v_C \tag{8.26}$$

the output equation can be written as

$$\begin{Bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \cdot u \tag{8.27}$$

Let us compare the state vectors of Examples 8.1 and 8.2, which are

$$\begin{aligned} \{x\} &= \{x_1 \quad x_2\}^T = \{q \quad \dot{q}\}^T; \\ \{\bar{x}\} &= \{\bar{x}_1 \quad \bar{x}_2\}^T = \{i \quad v_C\}^T \end{aligned} \tag{8.28}$$

The known relationships between the electrical variables, $q = C \cdot v_C$; $\dot{q} = i$, lead to the following transformation between the two sets of state variables:

$$\begin{Bmatrix} q \\ \dot{q} \end{Bmatrix} = \begin{bmatrix} 0 & C \\ 1 & 0 \end{bmatrix} \cdot \begin{Bmatrix} i \\ v_C \end{Bmatrix} \tag{8.29}$$

Eq. (8.29) indicates the two state vectors are linearly related, as illustrated in the generic Eq. (8.7), which showed how a state vector can be obtained by multiplying an existing state vector and a nonsingular square matrix. ■

Similar to the transfer function approach of Chapter 7, the subsequent material of this chapter is structured mainly in two segments: one is dedicated to deriving state-space models and the other one to calculating the forced response from a state-space model.

8.2 STATE-SPACE MODEL FORMULATION

The state-space models can be derived from existing time-domain models or from other models, such as the zero-pole-gain and the transfer function models. Obtaining state-space models can be pursued either analytically or by means of MATLAB, as discussed in the following sections.

8.2.1 State-Space Model From the Time-Domain Mathematical Model

The analytical procedures for state-space model derivation presented here apply to systems without input time derivatives, systems with input time derivatives, and nonlinear systems.

Dynamic Systems Without Input Time Derivative

When the differential equations that constitute the mathematical model of a dynamic system contain no time derivatives of the defined input, the procedure of selecting the state variables is rather straightforward, as will be discussed in the following. The cases of SISO and MIMO systems are analyzed separately.

SISO Systems

SISO linear, time-invariant dynamic systems that do not include time derivatives of their input are described by the differential equation

$$a_n \cdot \overset{(n)}{y}(t) + a_{n-1} \cdot \overset{(n-1)}{y}(t) + \ldots + a_1 \cdot \dot{y}(t) + a_0 \cdot y(t) = u(t)$$

or (8.30)

$$\overset{(n)}{y}(t) = -\frac{a_0}{a_n} \cdot y(t) - \frac{a_1}{a_n} \cdot \dot{y}(t) - \ldots - \frac{a_{n-1}}{a_n} \cdot \overset{(n-1)}{y}(t) + \frac{1}{a_n} \cdot u(t)$$

A rule of thumb is that the number of state variables is equal to the order of the differential equation representing the mathematical model of a dynamic system, provided a_0 and a_n of Eq. (8.30) are nonzero (the reader is encouraged to check the rationale for this assertion). This system is defined by a differential equation of order n and, therefore, requires n state variables, which can be selected as

$$x_1 = y, x_2 = \dot{y}, \ldots, x_{n-1} = \overset{(n-2)}{y}, x_n = \overset{(n-1)}{y} \tag{8.31}$$

The following relationships are therefore valid:

$$\dot{x}_1 = x_2, \dot{x}_2 = x_3, \ldots, \dot{x}_{n-1} = x_n \tag{8.32}$$

Eqs. (8.30) and (8.32) are collected into the matrix equation

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ldots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \ldots & -\frac{a_{n-1}}{a_n} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ \ldots \\ x_{n-1} \\ x_n \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ \frac{1}{a_n} \end{Bmatrix} \cdot u \tag{8.33}$$

which represents the state equation. $[A]$ is the square matrix multiplying the state vector, and $[B]$ is actually a column vector multiplying the scalar u. The output equation is determined by taking into account that the differential Eq. (8.30) has one unknown, y, which can be considered the output function, but $y = x_1$. As a consequence, the first Eq. (8.31) is written as

$$y = \{1 \quad 0 \quad \ldots \quad 0 \quad 0\} \cdot \{x_1 \quad x_2 \quad \ldots \quad x_{n-1} \quad x_n\}^T + 0 \cdot u \tag{8.34}$$

and, therefore, $[C]$ is the row vector, and $[D]$ is the zero scalar.

Examples 8.1 and 8.2 actually derive state-space models of dynamic systems without input time derivatives. Let us analyze another example from the same category.

Example 8.3

The microresonator sketched in Figure 8.2(a) is actuated electrostatically by a comb drive. The shuttle mass is supported by two elastic beams, and there is viscous damping between the shuttle mass and the substrate. Use the lumped-parameter model sketched in Figure 8.2(b) to determine the mathematical model of this mechanical system and derive a state-space model from it. The damping coefficient is c, one beam's spring constant is k, the shuttle mass is m, and the electrostatic actuation force is f.

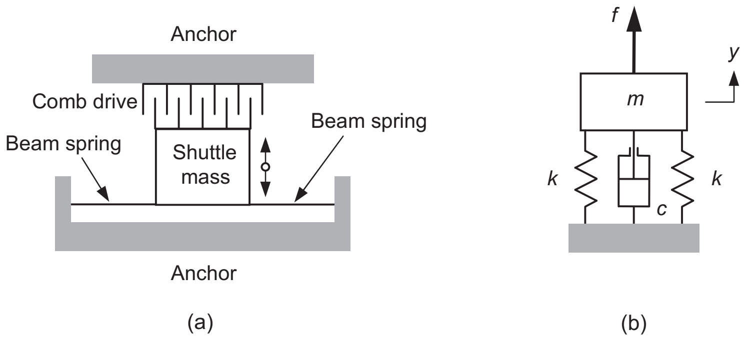

FIGURE 8.2

Microresonator Supported on Beam Springs, With Viscous Damping and Electrostatic Actuation: (a) Physical Model; (b) Lumped-Parameter Model.

Solution

Without including the free-body diagram, it can simply be shown using Newton's second law of motion that the equation of motion for the lumped-parameter model of Figure 8.2(b) is

$$m \cdot \ddot{y} + c \cdot \dot{y} + 2k \cdot y = f \tag{8.35}$$

which is a second-order differential equation with the following coefficients and forcing: $a_2 = m$, $a_1 = c$, $a_0 = 2k$, and $u = f$. Based on these parameter assignments and according to Eqs. (8.33) and (8.34), the state-space matrices are

$$[A] = \begin{bmatrix} 0 & 1 \\ -\dfrac{a_0}{a_2} & -\dfrac{a_1}{a_2} \end{bmatrix}; \quad [B] = \begin{Bmatrix} 0 \\ \dfrac{1}{a_2} \end{Bmatrix}; \quad [C] = \{1 \quad 0\}; \quad [D] = 0 \tag{8.36}$$

As a consequence, the generic state Eq. (8.33) and output Eq. (8.34) become

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{2k}{m} & -\dfrac{c}{m} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \dfrac{1}{m} \end{Bmatrix} \cdot u; \quad y = \{1 \quad 0\} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + 0 \cdot u \tag{8.37}$$

which form the state-space model of the micromechanical system of Figure 8.2(a).

MIMO Systems

The companion website Chapter 8 presents a formal procedure that can be used to identify state variables and to derive state-space models for MIMO systems with no input time derivative. A simplified version of that methodology is presented here, based on the following particular application. Assume the mathematical model of a MIMO system consists of two differential equations, one of second order and the other of first order. As discussed for SISO systems, the second-order differential equation requires two state variables, whereas for the first-order differential equation, one state variable is necessary, so three state variables are needed in total. A valid choice of the state variables, as seen in the examples that have been studied thus far, is the time-domain variable and its first time derivative for a second-order differential equation, whereas for a first-order differential equation, the sole state variable can be the time-domain variable of that equation. Let us analyze another example and derive its state-space model.

Example 8.4

Consider the mechanical system of Figure 8.3, where two forces f_1 and f_3 constitute the system input. Determine a state-space model for this system when the output vector is formed of the displacements y_1, y_2, and y_3.

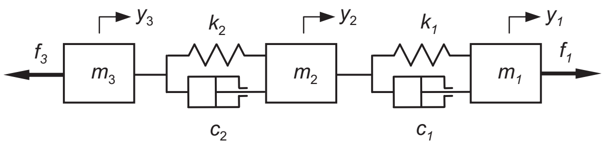

FIGURE 8.3

MIMO Translatory Mechanical System.

Solution

The following mathematical model can be obtained using either Newton's second law of motion or Lagrange's equations:

$$\begin{cases} m_1 \cdot \ddot{y}_1 = f_1 - c_1 \cdot \left(\dot{y}_1 - \dot{y}_2\right) - k_1 \cdot (y_1 - y_2) \\ m_2 \cdot \ddot{y}_2 = -c_1 \cdot \left(\dot{y}_2 - \dot{y}_1\right) - k_1 \cdot (y_2 - y_1) - c_2 \cdot \left(\dot{y}_2 - \dot{y}_3\right) - k_2 \cdot (y_2 - y_3) \\ m_3 \cdot \ddot{y}_3 = -f_3 - c_2 \cdot \left(\dot{y}_3 - \dot{y}_2\right) - k_2 \cdot (y_3 - y_2) \end{cases} \tag{8.38}$$

The mathematical model of this mechanical system consists of three second-order differential equations. As a consequence, we need $3 \times 2 = 6$ state variables, which are selected as

$$x_1 = y_1; \;\; x_2 = \dot{y}_1; \;\; x_3 = y_2; \;\; x_4 = \dot{y}_2; \;\; x_5 = y_3; \;\; x_6 = \dot{y}_3 \tag{8.39}$$

Eq. (8.39) indicates the following state variable connections:

$$\dot{x}_1 = x_2; \quad \dot{x}_3 = x_4; \quad \dot{x}_5 = x_6 \tag{8.40}$$

Defining the input vector as $\{u\} = \{u_1 \; u_2\}^T = \{f_1 \; f_3\}^T$, Eq. (8.38) can be written as

$$\begin{cases} \ddot{y}_1 = -\dfrac{k_1}{m_1}\cdot y_1 - \dfrac{c_1}{m_1}\cdot \dot{y}_1 + \dfrac{k_1}{m_1}\cdot y_2 + \dfrac{c_1}{m_1}\cdot \dot{y}_2 + \dfrac{1}{m_1}\cdot u_1 \\ \ddot{y}_2 = \dfrac{k_1}{m_2}\cdot y_1 + \dfrac{c_1}{m_2}\cdot \dot{y}_1 - \dfrac{k_1+k_2}{m_2}\cdot y_2 - \dfrac{c_1+c_2}{m_2}\cdot \dot{y}_2 + \dfrac{k_2}{m_2}\cdot y_3 + \dfrac{c_2}{m_2}\cdot \dot{y}_3 \\ \ddot{y}_3 = \dfrac{k_2}{m_3}\cdot y_2 + \dfrac{c_2}{m_3}\cdot \dot{y}_2 - \dfrac{k_2}{m_3}\cdot y_3 - \dfrac{c_2}{m_3}\cdot \dot{y}_3 - \dfrac{1}{m_3}\cdot u_2 \end{cases} \tag{8.41}$$

Collecting Eqs. (8.40) and (8.41) into matrix form results in the state equation

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\dfrac{k_1}{m_1} & -\dfrac{c_1}{m_1} & \dfrac{k_1}{m_1} & \dfrac{c_1}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \dfrac{k_1}{m_2} & \dfrac{c_1}{m_2} & -\dfrac{k_1+k_2}{m_2} & -\dfrac{c_1+c_2}{m_2} & \dfrac{k_2}{m_2} & \dfrac{c_2}{m_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{k_2}{m_3} & \dfrac{c_2}{m_3} & -\dfrac{k_2}{m_3} & -\dfrac{c_2}{m_3} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ \dfrac{1}{m_1} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\dfrac{1}{m_3} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \tag{8.42}$$

The matrix multiplying the state vector in Eq. (8.42) is $[A]$ and the one multiplying the input vector is $[B]$. The output vector $\{y\} = \{y_1\ y_2\ y_3\}^T$ is connected to the state and input vectors as in the first, third, and fifth Eq. (8.39)—they are collected in the output matrix equation:

$$\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \tag{8.43}$$

The matrix multiplying the state vector in Eq. (8.43) is $[C]$, and the zero matrix multiplying the input vector in the same equation is $[D]$. ■

Dynamic Systems With Input Time Derivative

In situations where the differential equations corresponding to a dynamical model include time derivatives of the defined input, the procedure used when there were no input time derivatives cannot be applied directly because a state-space model does not accommodate time derivatives of the input in its standard state and output equations. It is, however, possible to determine state-space models using an additional function or vector, as shown in the following. Only coverage of SISO systems is provided here, but the companion website Chapter 8 includes the study of MIMO systems.

Let us consider a SISO dynamic system whose mathematical model consists of the following differential equation:

$$\begin{aligned} & a_n \cdot \overset{(n)}{y}(t) + a_{n-1} \cdot \overset{(n-1)}{y}(t) + \ldots + a_1 \cdot \dot{y}(t) + a_0 \cdot y(t) \\ & = b_q \cdot \overset{(q)}{u}(t) + b_{q-1} \cdot \overset{(q-1)}{u}(t) + \ldots + b_1 \cdot \dot{u}(t) + b_0 \cdot u(t) \end{aligned} \tag{8.44}$$

The state-space variable choice made for SISO systems without input time derivatives cannot be applied here, simply because, with u being defined as (or being required to be) the input, the standard state-space model cannot have additional terms with time derivatives of u on the right-hand side of Eq. (8.44). Without loss of generality, let us assume that $n > q$. Application of the Laplace transform with zero initial conditions to Eq. (8.44) results in the following transfer function:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_q \cdot s^q + b_{q-1} \cdot s^{q-1} + \ldots + b_1 \cdot s + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \ldots + a_1 \cdot s + a_0} \tag{8.45}$$

A function $Z(s)$ is introduced in $G(s)$ of Eq. (8.45) as

$$\begin{aligned} G(s) = \frac{Z(s)}{U(s)} \cdot \frac{Y(s)}{Z(s)} = & \frac{1}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \ldots + a_1 \cdot s + a_0} \\ & \cdot \left(b_q \cdot s^q + b_{q-1} \cdot s^{q-1} + \ldots + b_1 \cdot s + b_0\right) \end{aligned} \tag{8.46}$$

The following relationships can be written from Eq. (8.46):

$$\begin{cases} \dfrac{Z(s)}{U(s)} = \dfrac{1}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \ldots + a_1 \cdot s + a_0} \\ \dfrac{Y(s)}{Z(s)} = b_q \cdot s^q + b_{q-1} \cdot s^{q-1} + \ldots + b_1 \cdot s + b_0 \end{cases} \tag{8.47}$$

Cross multiplication in Eq. (8.47) leads to

$$\begin{cases} \left(a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \ldots + a_1 \cdot s + a_0\right) \cdot Z(s) = U(s) \\ Y(s) = \left(b_q \cdot s^q + b_{q-1} \cdot s^{q-1} + \ldots + b_1 \cdot s + b_0\right) \cdot Z(s) \end{cases} \tag{8.48}$$

The inverse Laplace transform is applied to Eq. (8.48), which results in

$$\begin{cases} a_n \cdot \overset{(n)}{z}(t) + a_{n-1} \cdot \overset{(n-1)}{z}(t) + \ldots + a_1 \cdot \dot{z}(t) + a_0 \cdot z(t) = u(t) \\ y(t) = b_q \cdot \overset{(q)}{z}(t) + b_{q-1} \cdot \overset{(q-1)}{z}(t) + \ldots + b_1 \dot{z}(t) + b_0 \cdot z(t) \end{cases} \tag{8.49}$$

The first of the two Eq. (8.49) is now an equation in $z(t)$ that contains no time derivatives of the input $u(t)$ and, therefore, can be modeled using the standard procedure, which has been detailed in the previous section. This system needs n state variables that are

$$x_1 = z, x_2 = \dot{z}, \ldots, x_{n-1} = \overset{(n-2)}{z}, x_n = \overset{(n-1)}{z} \tag{8.50}$$

Finding the state equation is done exactly as shown previously and is not detailed here. To determine the output equation, the second Eq. (8.49) is written as

$$y = b_0 \cdot x_1 + b_1 \cdot x_2 + \ldots + b_{q-1} \cdot x_q + b_q \cdot x_{q+1} \tag{8.51}$$

which is written in vector-matrix form as

$$y = \{\, b_0 \quad b_1 \quad \ldots \quad b_q \quad \ldots \quad 0 \quad 0 \,\} \cdot \{\, x_1 \quad x_2 \quad \ldots \quad x_m \quad \ldots \quad x_{n-1} \quad x_n \,\}^T + 0 \cdot u \tag{8.52}$$

Note that for a SISO system with time derivatives of the input the numerator of the transfer function always contains terms in s; this problem is oftentimes referred to as *numerator dynamics*.

Example 8.5

Find a state-space representation for the electrical circuit of Figure 8.4 by using one state variable only. Consider that the input is the voltage v_i and the output is the voltage v_o.

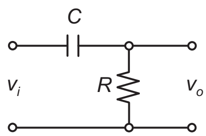

FIGURE 8.4

Capacitive—Resistive Electrical System.

Solution

The time-domain model can be obtained using KVL to express the input and output voltage. The reader is encouraged to check that the following differential equation describes the electrical system of Figure 8.4:

$$\dot{v}_o + \frac{1}{R \cdot C} \cdot v_o = \dot{v}_i \ \text{ or } \ R \cdot C \cdot \dot{v}_o + v_o = R \cdot C \cdot \dot{v}_i \tag{8.53}$$

which indicates that the time derivative of v_i is the input. As a consequence, the procedure described in this section needs to be utilized to derive a state-space model. The transfer function can be obtained applying the Laplace transform to Eq. (8.53) with zero initial conditions:

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{R \cdot C \cdot s}{R \cdot C \cdot s + 1} \tag{8.54}$$

The ratio of Eq. (8.54) is written by means of the intermediate function $Z(s)$ as

$$\frac{V_o(s)}{V_i(s)} = \frac{Z(s)}{V_i(s)} \cdot \frac{V_o(s)}{Z(s)} = \frac{1}{R \cdot C \cdot s + 1} \cdot (R \cdot C \cdot s) \tag{8.55}$$

with

$$\frac{Z(s)}{V_i(s)} = \frac{1}{R \cdot C \cdot s + 1} \ \text{ and } \ \frac{V_o(s)}{Z(s)} = R \cdot C \cdot s \tag{8.56}$$

Cross multiplication in the first Eq. (8.56), followed by application of the inverse Laplace transform, yields

$$R \cdot C \cdot \dot{z} + z = v_i \ \text{ or } \ \dot{z} = -\frac{1}{R \cdot C} \cdot z + \frac{1}{R \cdot C} \cdot v_i \tag{8.57}$$

If the state variable $x = z$ is chosen, Eq. (8.57) becomes the state equation

$$\dot{x} = -\frac{1}{R \cdot C} \cdot x + \frac{1}{R \cdot C} \cdot u \tag{8.58}$$

where the input voltage is the state-space input, $u = v_i$ and $A = -1/(R \cdot C)$, $B = 1/(R \cdot C)$.

Cross multiplication followed by inverse Laplace transformation are applied now to the second Eq. (8.56), which, combined with Eq. (8.57), yields the output equation

$$v_o = R \cdot C \cdot \dot{z} = -z + v_i \ \text{ or } \ y = -1 \cdot x + 1 \cdot u \tag{8.59}$$

where $y = v_o$ is the output, and $C = -1$, $D = 1$. Eqs. (8.58) and (8.59) form a state-space model for the electrical system of Figure 8.4. ■

Example 8.6

Derive a state-space model for the mechanical system shown in Figure 8.5(a), where the input is the displacement u, and the output is the displacement y.

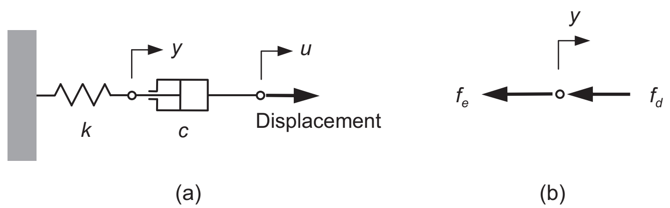

FIGURE 8.5

(a) Translatory Mechanical System With Displacement Input; (b) Free-Body Diagram.

Solution

Figure 8.5(b) shows the free-body diagram corresponding to the massless point defined by the coordinate y. Newton's second law of motion corresponding to this free-body diagram results in

$$0 = -f_d - f_e \quad \text{with } f_d = c \cdot (\dot{y} - \dot{u}); f_e = k \cdot y \tag{8.60}$$

where f_d and f_e are the damping and elastic forces. The following equation is obtained from Eq. (8.60):

$$c \cdot \dot{y} + k \cdot y = c \cdot \dot{u} \tag{8.61}$$

The mechanical system is SISO, and its mathematical model consists of a first-order differential equation. As a result, a single state variable is needed. Because the time derivative of the input is present, it is necessary to apply the Laplace transform to Eq. (8.61), which results in

$$\frac{Y(s)}{U(s)} = \frac{s}{s + \frac{k}{c}} \tag{8.62}$$

Using the intermediate function $Z(s)$ changes Eq. (8.62) to

$$\frac{Y(s)}{U(s)} = \frac{Z(s)}{U(s)} \cdot \frac{Y(s)}{Z(s)} = \frac{1}{s + \frac{k}{c}} \cdot s \tag{8.63}$$

which indicates the following selection needs to be made

$$\frac{Z(s)}{U(s)} = \frac{1}{s + \frac{k}{c}}; \quad \frac{Y(s)}{Z(s)} = s \tag{8.64}$$

Cross multiplication in the first Eq. (8.64) and application of the inverse Laplace transform to the resulting equation yields

$$c \cdot \dot{z} + k \cdot z = c \cdot u \tag{8.65}$$

Eq. (8.65) no longer contains input derivatives; therefore, we can choose $x = z$ as the state variable. As a consequence, Eq. (8.65) becomes

$$\dot{x} = -\frac{k}{c} \cdot x + u \tag{8.66}$$

which is the state equation with $A = -k/c$ and $B = 1$. Cross multiplication in the second Eq. (8.64) followed by inverse Laplace transformation results in

$$y = \dot{z} = \dot{x} \tag{8.67}$$

Combining Eqs. (8.66) and (8.67) yields

$$y = -\frac{k}{c} \cdot x + 1 \cdot u \tag{8.68}$$

which is the output equation with $C = -k/c$ and $D = 1$. ■

Nonlinear Systems

While it is almost always possible to derive a state-space model for a nonlinear dynamic system, finding a solution to it can be problematic. In such cases, the *linearization technique* can be utilized to derive a linear state−space model from the original, nonlinear one when the variables vary in small amounts around an *equilibrium* position. The companion website Chapter 8 analyzes the linearization of nonlinear nonhomogeneous state-space models, that is, models that contain a nonzero input vector, but here we analyze only homogeneous nonlinear systems.

Consider a two-DOF dynamic system defined by one second-order differential equation and one first-order differential equation. As previously discussed in this chapter, we use three state variables: two corresponding to the second-order differential equation, x_1 and x_2, and one for the first-order differential equation, x_3. Because the system has two DOF, the DOF variables are selected to be the output variables, y_1 and y_2. As a consequence, the following nonlinear state equations can be formulated:

$$\begin{cases} \dot{x}_1(t) = f_1(x_1, x_2, x_3) \\ \dot{x}_2(t) = f_2(x_1, x_2, x_3) \\ \dot{x}_3(t) = f_3(x_1, x_2, x_3) \end{cases} \tag{8.69}$$

where f_1, f_2, and f_3 are nonlinear functions of x_1, x_2, and x_3. Similarly, the output variables can be expressed as

$$\begin{cases} y_1(t) = h_1(x_1, x_2, x_3) \\ y_2(t) = h_2(x_1, x_2, x_3) \end{cases} \tag{8.70}$$

with h_1 and h_2 being nonlinear functions of x_1, x_2, and x_3. If small (linear) variations of the time-dependent variables and functions of Eq. (8.69) are considered (which is equivalent to using Taylor series expansions of the functions on the left-hand sides of Eq. (8.69) about the equilibrium point), the following equations result:

$$\begin{cases} \delta\dot{x}_1(t) = \left(\dfrac{\partial f_1}{\partial x_1}\right)_e \cdot \delta x_1(t) + \left(\dfrac{\partial f_1}{\partial x_2}\right)_e \cdot \delta x_2(t) + \left(\dfrac{\partial f_1}{\partial x_3}\right)_e \cdot \delta x_3(t) \\ \delta\dot{x}_2(t) = \left(\dfrac{\partial f_2}{\partial x_1}\right)_e \cdot \delta x_1(t) + \left(\dfrac{\partial f_2}{\partial x_2}\right)_e \cdot \delta x_2(t) + \left(\dfrac{\partial f_2}{\partial x_3}\right)_e \cdot \delta x_3(t) \\ \delta\dot{x}_3(t) = \left(\dfrac{\partial f_3}{\partial x_1}\right)_e \cdot \delta x_1(t) + \left(\dfrac{\partial f_3}{\partial x_2}\right)_e \cdot \delta x_2(t) + \left(\dfrac{\partial f_3}{\partial x_3}\right)_e \cdot \delta x_3(t) \end{cases} \tag{8.71}$$

where the variables preceded by the symbol δ indicate small variations of those variables, and the subscript e indicates the equilibrium point. Eq. (8.71) can be arranged into the following matrix equation:

$$\begin{Bmatrix} \delta\dot{x}_1(t) \\ \delta\dot{x}_2(t) \\ \delta\dot{x}_3(t) \end{Bmatrix} = \begin{bmatrix} \left(\dfrac{\partial f_1}{\partial x_1}\right)_e & \left(\dfrac{\partial f_1}{\partial x_2}\right)_e & \left(\dfrac{\partial f_1}{\partial x_3}\right)_e \\ \left(\dfrac{\partial f_2}{\partial x_1}\right)_e & \left(\dfrac{\partial f_2}{\partial x_2}\right)_e & \left(\dfrac{\partial f_2}{\partial x_3}\right)_e \\ \left(\dfrac{\partial f_3}{\partial x_1}\right)_e & \left(\dfrac{\partial f_3}{\partial x_2}\right)_e & \left(\dfrac{\partial f_3}{\partial x_3}\right)_e \end{bmatrix} \cdot \begin{Bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \end{Bmatrix} \tag{8.72}$$

Eq. (8.72) is the linearized version of the original, nonlinear state equation. The matrix $[A]$ is formed of the partial derivatives of f_1, f_2, f_3 in terms of x_1, x_2, x_3, the evaluation being made at the equilibrium point. The state vector is formed of small variations of the original state vector's components.

If small variations are now considered for the nonlinear output Eq. (8.70), the following equations are obtained:

$$\begin{cases} \delta y_1(t) = \left(\dfrac{\partial h_1}{\partial x_1}\right)_e \cdot \delta x_1(t) + \left(\dfrac{\partial h_1}{\partial x_2}\right)_e \cdot \delta x_2(t) + \left(\dfrac{\partial h_1}{\partial x_3}\right)_e \cdot \delta x_3(t) \\ \delta y_2(t) = \left(\dfrac{\partial h_2}{\partial x_1}\right)_e \cdot \delta x_1(t) + \left(\dfrac{\partial h_2}{\partial x_2}\right)_e \cdot \delta x_2(t) + \left(\dfrac{\partial h_2}{\partial x_3}\right)_e \cdot \delta x_3(t) \end{cases} \tag{8.73}$$

which can be written in vector-matrix form as

$$\begin{Bmatrix} \delta y_1(t) \\ \delta y_2(t) \end{Bmatrix} = \begin{bmatrix} \left(\dfrac{\partial h_1}{\partial x_1}\right)_e & \left(\dfrac{\partial h_1}{\partial x_2}\right)_e & \left(\dfrac{\partial h_1}{\partial x_3}\right)_e \\ \left(\dfrac{\partial h_2}{\partial x_1}\right)_e & \left(\dfrac{\partial h_2}{\partial x_2}\right)_e & \left(\dfrac{\partial h_2}{\partial x_3}\right)_e \end{bmatrix} \cdot \begin{Bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \end{Bmatrix} \tag{8.74}$$

Eq. (8.74) is the linearized form of the original, nonlinear output Eq. (8.70). The output vector is connected to the state vector through the matrix $[C]$.

Example 8.7

The lumped-parameter model of a microcantilever that vibrates in a vacuum is the mass-spring system shown in Figure 8.6. Considering that the spring constant is nonlinear such that the elastic force is $f_e = (1/3) \cdot y^3$, where y is the vertical displacement measured from the equilibrium position, find a state-space representation of the system and derive the corresponding linearized state-space model. Known are the lumped mass m and gravitational acceleration g.

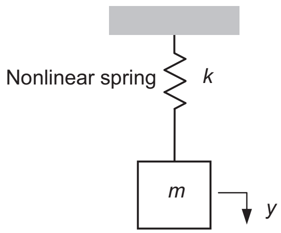

FIGURE 8.6

Lumped-Parameter Model of Microcantilever With Nonlinear Spring.

Solution

At static equilibrium, the gravity force is equal to the elastic force; therefore,

$$m \cdot g = \frac{1}{3} \cdot y_e^3 \text{ or } y_e = \sqrt[3]{3m \cdot g} \tag{8.75}$$

where y_e is the deformation of the spring at static equilibrium. By relocating the reference frame to the static equilibrium position defined by y_e of Eq. (8.75), the gravity action is removed, as known from dynamics theory, and the equation of motion is

$$m \cdot \ddot{y} = -\frac{1}{3} \cdot y^3 \tag{8.76}$$

where y is the distance measured from the equilibrium position to an arbitrary position. The following state variables are selected:

$$x_1 = y; \quad x_2 = \dot{y} \tag{8.77}$$

which enables formulating the equations

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = -\frac{1}{3m} \cdot x_1^3 \tag{8.78}$$

These are state equations, and the second one is nonlinear. The output equation is linear:

$$y = x_1 \tag{8.79}$$

Let us now linearize the state equation. If differentials in functions and in variables are considered, then the first Eq. (8.78), which is linear, can be written as

$$\delta\dot{x}_1 = \delta x_2 \tag{8.80}$$

The second Eq. (8.78) is reformulated based on the previous theory discussion as

$$\delta\dot{x}_2 = -\frac{1}{3m} \cdot \frac{\partial}{\partial x_1}\left(x_1^3\right)\bigg|_{x_1 = x_{1,e}} \cdot \delta x_1 = -\frac{1}{m} \cdot x_{1,e}^2 \cdot \delta x_1 = -\frac{1}{m} \cdot y_e^2 \cdot \delta x_1 \tag{8.81}$$

Because $x_1 = y$ and y_e is provided in Eq. (8.75), Eq. (8.81) becomes

$$\delta \dot{x}_2 = -\sqrt[3]{9g^2/m} \cdot \delta x_1 \tag{8.82}$$

Eqs. (8.80) and (8.82) can now be collected into the linearized state equation:

$$\left\{ \begin{array}{c} \delta \dot{x}_1 \\ \delta \dot{x}_2 \end{array} \right\} = \begin{bmatrix} 0 & 1 \\ -\sqrt[3]{9g^2/m} & 0 \end{bmatrix} \cdot \left\{ \begin{array}{c} \delta x_1 \\ \delta x_2 \end{array} \right\} \tag{8.83}$$

where the state variables are δx_1 and δx_2. The output equation is simply obtained by differentiating Eq. (8.79) as

$$\delta y = \delta x_1 \tag{8.84}$$

which is written in standard form as

$$\delta y = \{ 1 \quad 0 \} \cdot \left\{ \begin{array}{c} \delta x_1 \\ \delta x_2 \end{array} \right\} \tag{8.85}$$

The state Eq. (8.83) and output Eq. (8.85) form the linearized state-space model of the mechanical system of Figure 8.6. ■

8.2.2 State-Space Model From Other Models

The state-space models can be obtained from other models, such as transfer function or zero-pole-gain models, both analytically and by using specialized MATLAB commands. It is also possible to convert state-space models into other models, as discussed in this section.

Conversions Between Transfer Function and State-Space Models

Transformation of a Transfer Function Model Into a State-Space Model

The known transfer function of a SISO system can be converted analytically into a state-space model by following the steps indicated in the previous subsection, where the state-space model of a SISO system with input time derivatives is obtained using the transfer function and an intermediate function $Z(s)$. The companion website Chapter 8 covers the topic of analytic conversion for MIMO systems. MATLAB also enables converting between transfer function and state-space models for both SISO and MIMO systems, as discussed next.

A MATLAB state-space model or LTI (linear time invariant) object is defined by using the command `sys = ss (A, B, C, D)`, where `A, B, C`, and `D` are the state-space model defining matrices. For instance, if the command is issued

```
>> sys = ss ([1, 3; 2, 5], [1; 3], [-1, 0; 0, 5], [0; 1])
```

the following result will be generated:

```
a =                b =            c =                d =

        x1   x2          u1            x1   x2             u1

   x1   1    3      x1   1        y1   -1   0         y1   0

   x2   2    5      x2   3        y2   0    5         y2   1
```

which is an explicit way to mention the vectors that connect to any of the system's matrices. Combining this command with the `tf` MATLAB command it is possible to realize conversions between transfer function and state-space models. Note that to be able to realize MATLAB conversion, the transfer function should not be algebraic—it needs numerical values for its coefficients.

Example 8.8

Determine a state-space model by converting the transfer function model of the operational amplifier circuit sketched in Figure 7.9(a) of Example 7.6 by the analytical approach. Also utilize MATLAB to obtain a state-space model. Consider the following actual elements: a resistor $R_1 = 100\,\Omega$ for Z_1, a resistor $R_2 = 120\,\Omega$ for Z_2, a resistor $R_3 = 75\,\Omega$ for Z_3, and a capacitor $C = 2 \cdot 10^{-4}$ F for Z_4.

Solution

Eq. (7.43) gives the system's transfer function, which can be written as

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{Z_2 \cdot Z_4}{Z_1 \cdot Z_3} = \frac{Z(s)}{V_i(s)} \cdot \frac{V_o(s)}{Z(s)}$$

$$= \frac{1}{R_1 \cdot R_3 \cdot C \cdot s} \cdot R_2 \ \text{ or } \left\{ \begin{array}{l} \dfrac{Z(s)}{V_i(s)} = \dfrac{1}{R_1 \cdot R_3 \cdot C \cdot s}; \\ \dfrac{V_o(s)}{Z(s)} = R_2 \end{array} \right. \tag{8.86}$$

The four impedances of Eq. (8.86)—not to be confounded with the function Z—are $Z_1 = R_1$, $Z_2 = R_2$, $Z_3 = R_3$, and $Z_4 = 1/(C \cdot s)$. Cross multiplication in the first Eq. (8.86) and application of the inverse Laplace transformation to the resulting equation (with zero initial conditions) results in the following equation:

$$\dot{z} = \frac{1}{R_1 \cdot R_3 \cdot C} \cdot v_i \tag{8.87}$$

By selecting the state variable as $x = z$ and considering that the input is $u = v_i$ changes Eq. (8.87) to

$$\dot{x} = \frac{1}{R_1 \cdot R_3 \cdot C} \cdot u \tag{8.88}$$

which is the scalar form of the state equation with $A = 0$ and $B = 1/(R_1 \cdot R_3 \cdot C) = 0.667$. Cross multiplication and inverse Laplace transformation with zero initial conditions are also applied to the second Eq. (8.86), which results in

$$y = R_2 \cdot x \tag{8.89}$$

where $y = v_o$. Eq. (8.89) is the output equation with $C = R_2 = 120$ (C is the 1×1 [C] matrix of the output equation) and $D = 0$.

The following MATLAB code:

```
>> R1 = 100; R2 = 120; R3 = 75; C = 2e-4;
>> G = tf (R2, [R1*R3*C, 0]);
>> ss(G)
```

returns

```
a =
     x1
  x1  0
b =
     u1
  x1  8
c =
     x1
  y1  10
d =
     u1
  y1  0
```

which indicates that $B = 8$ and $C = 10$. While individually, the B and C values obtained by MATLAB differ from the analytical results, it should be taken into account that the transfer function of the analyzed system is calculated with Eq. (8.14) as:

$$G(s) = [C]\cdot(s\cdot[I])^{-1}\cdot[B] = \frac{C\cdot B}{s} = \frac{80}{s} \tag{8.90}$$

since $A = 0$ and $D = 0$. It can also be seen that the $C\cdot B = 80$ for both the analytical and the MATLAB values of B and C. Moreover, any two values of B and C whose product equals 80 would be a satisfactory solution to this problem, and this illustrates again that a state-space model is not unique.

■

Example 8.9

Use MATLAB to convert the transfer function matrix

$$[G(s)] = \begin{bmatrix} \dfrac{3}{s^2+s+1} & \dfrac{1}{s^2+s+1} \\ \dfrac{s}{s^2+s+1} & \dfrac{1}{s^2+s+1} \end{bmatrix}$$

into a state-space model.

Solution

The following MATLAB sequence solves this example:

```
>> s = tf ('s');
>> g = [3/(s^2 + s + 1), 1/ (s^2 + s + 1); s/( s^2 + s + 1), 1/( s^2 + s + 1)];
>> f = ss(g);
```

To identify (and store) the four state-space matrices, the following command can be added to this sequence:

```
>> [a,b,c,d] = ssdata(f)
```

which returns.

```
a =
          -1          -1          0           0
          1           0           0           0
          0           0           -1          -1
          0           0           1           0
b =
          2           0
          0           0
          0           2
          0           0
c =
          0           1.5         0           0.5
          0.5         0           0           0.5
d =
          0           0
          0           0
```

As it can be seen from the result, the state-space generated through conversion from the given transfer function uses four state variables (see the dimensions of [*A*]), two inputs, and two outputs (see the dimension of [*D*]), which is, obviously, just one possible solution from an infinite number of potential state-space solutions.

Transformation of a State-Space Model Into a Transfer Function Model

This subsection studies the topic of transforming a state-space model into the corresponding transfer function model. $\{Y(s)\}$ and $\{U(s)\}$, the Laplace transforms of the output and input vectors, are connected by means of transfer function matrix $[G(s)]$, as shown in Eq. (8.14). Since $[G(s)]$ is calculated based on the state-space model matrices $[A]$, $[B]$, $[C]$, and $[D]$, it follows that the respective calculation procedure enables transformation of a state-space model into a transfer function one for both SISO and MIMO systems. Let us study the following example.

Example 8.10

The following matrices define a state-space model:

$$[A] = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}; \quad [B] = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}; \quad [C] = \{2 \quad 1\}; \quad [D] = 0$$

Find the corresponding transfer function model using analytical calculation and also applying MATLAB conversion.

Solution

The dimensions of the state-space model matrices indicate this is a SISO system and, as a consequence, the transfer function is a scalar function $G(s)$. The definition Eq. (8.14) becomes

$$G(s) = [C] \cdot (s \cdot [I] - [A])^{-1} \cdot [B] \tag{8.91}$$

due to the fact that $D = 0$. The following MATLAB code:

```
>> syms s
>> A = [1, 0; 0, 2]; B = [0; 1]; C=[2, 1];
>> G = simplify(C*inv(s*eye(2)-A)*B)
```

returns the transfer function

$$G(s) = \frac{1}{s-2} \tag{8.92}$$

The following MATLAB code

```
>> A = [1, 0; 0, 2]; B = [0; 1]; C=[2, 1]; D=0;
>> S = ss(A,B,C,D);
>> tf(S)
```

returns the transfer function of Eq. (8.92).

Conversion Between Zero-Pole-Gain and State-Space Models

Conversion between transfer function and zero-pole-gain models is presented in Chapter 7. Similarly, MATLAB enables conversion between state-space and zero-pole-gain models. Assume that an LTI object `sys1` has been defined as a zero-pole-gain model; the MATLAB command

```
>> sys2 = ss(sys1)
```

converts the zero-pole-gain `sys1` model into another state-space model, labeled `sys2`. Conversely, when a state-space model, named `sys2`, is available, it can be transformed into a zero-pole-gain model, named `sys1`, by means of the MATLAB command as follows:

```
>> sys1 = zpk(sys2)
```

Example 8.11

A dynamic system's zero-pole-gain model is defined by a gain of 2, its zeroes are 1 (double) and 3, and its poles are 0 and −1 (triple). Determine a state-space model corresponding to the given zero-pole-gain model then confirm that the state-space model can be converted back to the original zero-pole-gain model.

Solution

The following MATLAB sequence solves this example:

```
>> sys1 = zpk([1,1,3],[-1,-1,-1,0],2)
Zero/pole/gain:
2 (s-1)^2 (s-3)
---------------
 s (s+1)^3
>> sys2 = ss(sys1)


a =
          x1        x2          x3          x4
          x1        0           -1.414      -2          1
```

```
          x2     0       -1       -2.828     1.414
          x3     0       0        -1         2
          x4     0       0        0          -1
b =
          u1
          x1     0
          x2     0
          x3     0
          x4     4
c =
                 x1      x2       x3         x4
          y1     -0.5    -0.7071  -1         0.5
d =
          u1
          y1     0

   Continuous-time model.
   >> sys3=zpk(sys2)
   Zero/pole/gain:
   2 (s-3) (s-1)^2
   ---------------
    s (s+1)^3
```

It can be seen that the state-space model rendered by MATLAB (one of the many possible models) is the one of a SISO system (as indicated by the dimensions of the matrix [*D*]), which uses four state variables (see the dimensions of matrix [*A*]). By applying the zero-pole-gain command to the state-space model, the original zero-pole-gain object is retrieved. ■

8.3 STATE-SPACE MODEL AND THE TIME-DOMAIN RESPONSE

Analytical, MATLAB, and Simulink methods can be used to determine the time (forced) response of dynamic systems by means of state-space models, which is the subject of this section.

8.3.1 Analytical Approach: The State-Transition Matrix Method

As briefly mentioned at the beginning of this chapter, the time response of state-space-modeled dynamic systems can be evaluated by an algorithm that consists of combined calculations in the Laplace and time domains. In essence, solving for the output $\{y\}$ based on a given input $\{u\}$ and a selected state vector $\{x\}$ can be performed as indicated in Figure 8.7. The state vector $\{x\}$ is first determined through integration from the state equation, then substituted into the output equation, which yields the output vector $\{y\}$ as a function of the input vector $\{u\}$, the state vector $\{x\}$, and the state-space model matrices $[A]$, $[B]$, $[C]$, and $[D]$. Two subcases are studied next: The first one analyzes the homogeneous state-space model (with no forcing

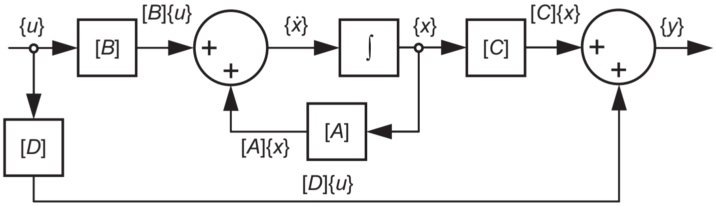

FIGURE 8.7

Block Diagram of Operations in a State-Space Model.

$\{u(t)\}$), and the other one focuses on the nonhomogeneous state-space model where $\{u(t)\} \neq 0$. Both categories consider nonzero initial conditions. The *state-transition matrix method* is utilized, as explained next.

Homogeneous State-Space Model

The homogeneous case corresponds to the absence of an input vector, whereby the system response is caused by initial conditions only. The state and output equations are given in Eq. (8.6). The aim of this method is to express the state vector at any time instant in terms of the initial state vector as

$$\{x(t)\} = [\varphi(t)] \cdot \{x(0)\} \tag{8.93}$$

where $[\varphi(t)]$ is the *state- transition matrix*. There are two methods of calculating the state-transition matrix: the *Laplace transform method*, which is discussed next, and the *matrix exponential method* that is presented in the companion website Chapter 8.

By applying the Laplace transform to the state equation, the first Eq. (8.6), and considering a nonzero initial state vector $\{x(0)\}$, the following relationship is obtained:

$$\{X(s)\} = (s \cdot [I] - [A])^{-1} \cdot \{x(0)\} \tag{8.94}$$

whose inverse Laplace transform yields the time-domain state vector:

$$\{x(t)\} = \mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1}\right] \cdot \{x(0)\} \tag{8.95}$$

By comparing Eqs. (8.93) and (8.95), it follows that the matrix connecting $\{x(t)\}$ to $\{x(0)\}$ is the state-transition matrix, which is therefore calculated as

$$[\varphi(t)] = \mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1}\right] \tag{8.96}$$

The output equation, which is the second Eq. (8.6), becomes

$$\{y(t)\} = [C] \cdot [\varphi(t)] \cdot \{x(0)\} = [C] \cdot \mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1}\right] \cdot \{x(0)\} \tag{8.97}$$

Example 8.12

The homogeneous state-space model of a dynamic system has the matrices $[A] = \begin{bmatrix} 0 & 2 & 0 \\ -8 & -1 & 8 \\ 0 & 0 & 1 \end{bmatrix}$; $[C] = \{1 \quad 0 \quad 1\}$.

Determine the system's response $y(t)$ and plot it with respect to time if the following initial condition is used: $\{x(0)\} = \{0\ 1\ 0\}^T$.

Solution

The following MATLAB commands

```
>> a = [0,2,0;-8,-1,8;0,0,1];
>> c = [1,0,1];
>> syms s
>> c*ilaplace(inv(s*eye(3)-a))*[0;1;0]
```

which are based on the analytical Eq. (8.97), yield the following approximate output:

$$y(t) = 0.5 \cdot e^{-0.5t} \cdot \sin(4t) \tag{8.98}$$

The response curve is plotted in Figure 8.8.

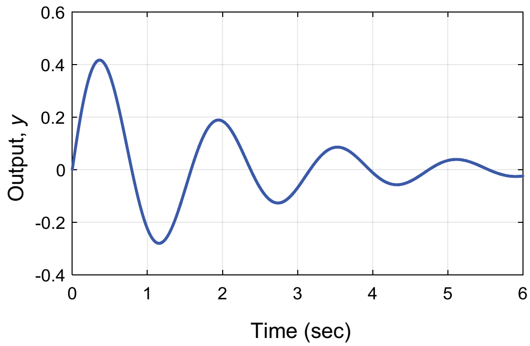

FIGURE 8.8

Time Response of a Homogeneous State-Space Model.

Example 8.13

Use the state-space approach to find the response of the liquid system shown in Figure 7.11 of the solved Example 7.8 by considering that there is no input flow rate. Instead, there is an initial value of the fluid head in the left tank, $h_1(0)$, which acts as an initial condition. Considering that the heads h_1 and h_2 are the output variables, plot $h_1(t)$ and $h_2(t)$ for $R_1 = 20\ \mathrm{s/m^2}$, $R_2 = 25\ \mathrm{s/m^2}$, $C_1 = 1\ \mathrm{m^2}$, $C_2 = 2\ \mathrm{m^2}$, and $h_1(0) = 0.05$ m.

Solution

For the case of the free response, the differential equations are derived using the definitions of hydraulic capacitances, Eq. (5.12), and resistances, Eq. (5.23):

$$C_1 \cdot \frac{dh_1}{dt} = -q; \;\; C_2 \cdot \frac{dh_2}{dt} = q - q_o; \;\; R_1 \cdot q = h_1 - h_2; \;\; R_2 \cdot q_o = h_2 \tag{8.99}$$

The flow rates q and q_o are substituted from the last two Eq. (8.99) into the first two Eq. (8.99), which yields:

$$\begin{cases} C_1 \cdot \dfrac{dh_1}{dt} + \dfrac{1}{R_1} \cdot h_1 - \dfrac{1}{R_1} \cdot h_2 = 0 \\ C_2 \cdot \dfrac{dh_2}{dt} - \dfrac{1}{R_1} \cdot h_1 + \left(\dfrac{1}{R_1} + \dfrac{1}{R_2}\right) \cdot h_2 = 0 \end{cases} \tag{8.100}$$

Two state variables are needed for this two-DOF first-order system, and they can be selected as

$$x_1 = h_1; \;\; x_2 = h_2 \tag{8.101}$$

By using these state variables in conjunction with Eq. (8.100), the following state equation is obtained:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} -\dfrac{1}{R_1 \cdot C_1} & \dfrac{1}{R_1 \cdot C_1} \\ \dfrac{1}{R_1 \cdot C_2} & -\dfrac{1}{C_2} \cdot \left(\dfrac{1}{R_1} + \dfrac{1}{R_2}\right) \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \tag{8.102}$$

Matrix $[A]$ of the state-space model is the one connecting the two vectors in Eq. (8.102). The output equation is obtained based on Eq. (8.101):

$$\begin{Bmatrix} h_1 \\ h_2 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \tag{8.103}$$

Matrix $[C]$ of the state-space model is the identity matrix $[I]$; therefore, $\{x(0)\} = \{y(0)\} = \{h_1(0)\;0\}^T$. Using the parameters of this example, Eq. (8.97) yields the following time-domain output:

$$\begin{cases} h_1 = 0.05 \cdot [\cos h(0.035 \cdot t) - 0.07 \sin h(0.035 \cdot t)] \cdot e^{-0.047 \cdot t} \\ h_2 = 0.035 \cdot \sinh(0.035 \cdot t) e^{-0.047 \cdot t} \end{cases} \tag{8.104}$$

Figure 8.9 contains the plots of the two response components. For $t \to \infty$, the heads become $h_1(\infty) = 0$ and $h_2(\infty) = 0$. ■

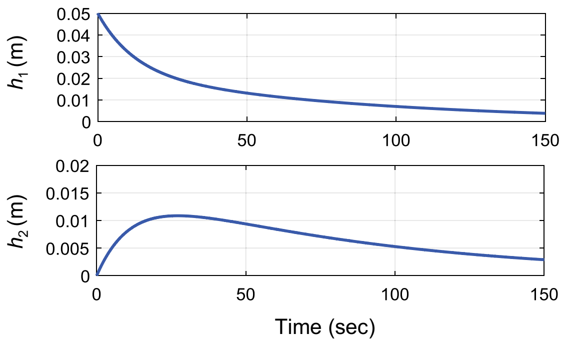

FIGURE 8.9

Time-Response Curves of Hydraulic Heads.

Nonhomogeneous State-Space Model

The nonhomogeneous case implies intervention of the input (forcing) vector $\{u(t)\}$; therefore, the state and output equations are the ones originally expressed in Eq. (8.6). Application of the Laplace transform to the state Eq. (8.6), considering there is a nonzero initial state vector $\{x(0)\}$, results in the following Laplace-domain equation:

$$\{X(s)\} = (s \cdot [I] - [A])^{-1} \cdot \{x(0)\} + (s \cdot [I] - [A])^{-1} \cdot [B] \cdot \{U(s)\} \tag{8.105}$$

The inverse Laplace transform is now applied to Eq. (8.105), which leads to

$$\{x(t)\} = [\varphi(t)] \cdot \{x(0)\} + \mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1} \cdot [B] \cdot \{U(s)\}\right] \tag{8.106}$$

According to the convolution theorem, which is studied in Chapter 6, the second term on the right-hand side of Eq. (8.106) can be calculated as

$$\begin{aligned}\mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1} \cdot [B] \cdot \{U(s)\}\right] &= \mathcal{L}^{-1}\left[(s \cdot [I] - [A])^{-1}\right] * \mathcal{L}^{-1}[[B] \cdot \{U(s)\}] \\ &= \int_0^t \left[\varphi(t-\tau)\right] \cdot [B] \cdot \{u(\tau)\} d\tau\end{aligned} \tag{8.107}$$

As a consequence, Eq. (8.106) becomes

$$\{x(t)\} = [\varphi(t)] \cdot \{x(0)\} + \int_0^t \left[\varphi(t-\tau)\right] \cdot [B] \cdot \{u(\tau)\} d\tau \tag{8.108}$$

where $[\varphi(t)]$ is calculated by means of Eq. (8.96). The output vector is found as

$$\{y(t)\} = [C] \cdot \left(\left[\varphi(t)\right] \cdot \{x(0)\} + \int_0^t \left[\varphi(t-\tau)\right] \cdot [B] \cdot \{u(\tau)\} d\tau\right) + [D] \cdot \{u(t)\} \tag{8.109}$$

Example 8.14

A dynamic system is described in state-space form by the matrices

$$[A] = \begin{bmatrix} 0 & 1 \\ -60 & -5 \end{bmatrix}; \quad [B] = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}; \quad [C] = \{1 \quad 0\}; \quad [D] = 0$$

A step input $u = 3$ is applied to the system with zero initial conditions. Find the system response using the state-transition matrix approach and plot it against time.

Solution

The matrices dimensions show that the system is SISO and there are two state variables. For the particular case of this example, the output is expressed from Eq. (8.109) as

$$\{y(t)\} = [C] \cdot \int_0^t \varphi(t-\tau) \cdot [B] \cdot u(\tau) d\tau \tag{8.110}$$

where $u(\tau) = 3$. The state-transition matrix $[\varphi(t)]$ is first computed by using the definition Eq. (8.96) as

$$[\varphi(t)] \simeq \begin{bmatrix} [\cos(7.3 \cdot t) + 0.34 \cdot \sin(7.3 \cdot t)] & 0.14 \cdot \sin(7.3 \cdot t) \\ -8.18 \cdot \sin(7.3 \cdot t) & [\cos(7.3 \cdot t) - 0.34 \cdot \sin(7.3 \cdot t)] \end{bmatrix} \cdot e^{-2.5 \cdot t} \tag{8.111}$$

The variable $t—\tau$ is used instead of the variable t in Eq. (8.111), then the necessary operations are performed in Eq. (8.110), which yields the following time-domain response:

$$y(t) = 0.005 - 0.017 \cdot [\sin(7.3 \cdot t) + 2.933 \cdot \cos(7.3 \cdot t)] \cdot e^{-2.5 \cdot t} \tag{8.112}$$

Figure 8.10 displays the output as a function of time. ■

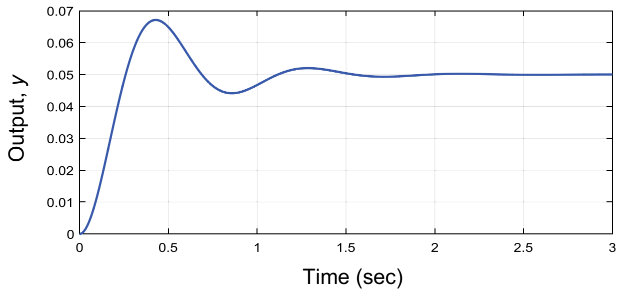

FIGURE 8.10

Time Response of Nonhomogeneous State-Space Model.

8.3.2 MATLAB Approach

This section presents the application of built-in MATLAB functions that use state-space modeling to calculate the forced response (including nonzero initial conditions) of dynamic systems and to plot it in terms of time. The free response with nonzero initial conditions and the forced response are studied next.

Free Response With Nonzero Initial Conditions

With MATLAB, it is possible to directly model the free response of a state-space dynamic system when the initial conditions are different from zero. The basic command is as follows:

```
>> initial (sys, x0)
```

where x_0 is the initial state vector, and `sys` is a state-space model. This command directly plots the state-space output y in terms of time.

Example 8.15

The free-response state-space model of a dynamic system is represented by the matrices [A] and [C] and the initial state vector x_0, which are defined as

$$[A] = \begin{bmatrix} -1 & -1.5 \\ 1.5 & 0.1 \end{bmatrix}; \quad [C] = \{4 \quad 10\}; \quad \{x_0\} = \begin{Bmatrix} 10 \\ 1 \end{Bmatrix}$$

Use MATLAB to determine the time response and plot it in terms of time.

Solution

The following MATLAB command sequence

```
>> a = [-1, -1.5; 1.5, 0.1];
>> c = [4, 10];
>> x0 = [10; 1];
>> sys = ss (a, [], c, []);
>> initial (sys, x0)
```

is used to model the free response of this specific state-space model, and the result is plotted in Figure 8.11.

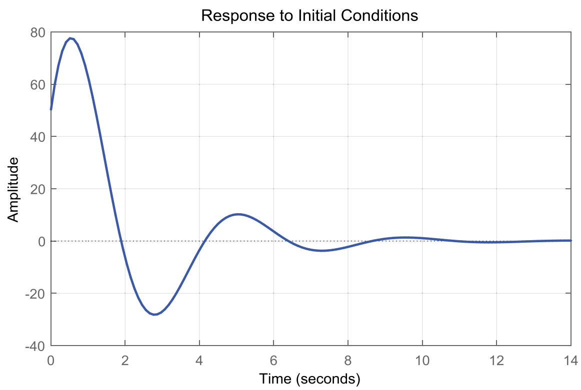

FIGURE 8.11

Example Plot of Free Response With Nonzero Initial Conditions.

Several previously defined state-space models can be plotted on the same graph and time interval and specified with the command

```
>> initial (sys1, sys2,…,sysn, x0, t)
```

where `t` is a time interval. When using the command.

```
>> [y,t,x] = initial (sys1, sys2,…,sysn, x0, t)
```

no plot is returned but the output matrix and state matrix are formed, each having as many rows as time increments. The number of columns in y is equal to the number of inputs, and the number of columns in x is equal to the number of state variables.

Subsequent plotting is possible with a `plot` command. Instead of `initial`, `initialplot` can be used with the same specifications.

Forced Response

To find and plot the forced time response, the state-space models in MATLAB employ the same predefined functions as the transfer function models, `step`, `impulse`, and `lsim`. For instance, if a state-space model is defined as

```
>> sys = ss ([-1, -1; 1, 0], [1; 0], [1, 0; 0, 2], [0; 1]);
```

the command `step(sys)` produces the plots shown in Figure 8.12. Because the matrices of this particular example define a one-input, two-output model, two plots result, each corresponding to a unit step input.

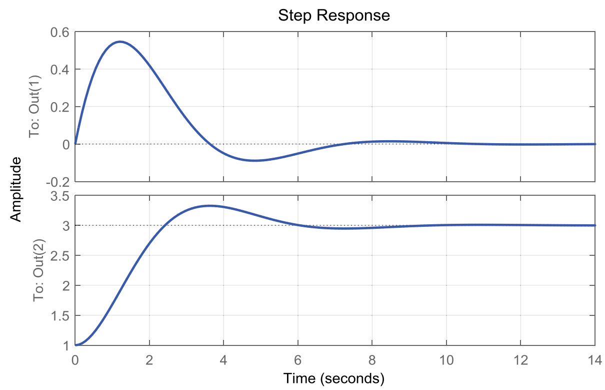

FIGURE 8.12

Response of a Two-Output State-Space Model to Unit-Step Input.

Example 8.16

Consider the mechanical microsystem shown in Figure 8.13, which is formed of three identical serpentine springs, each having a spring constant of $k = 10$ N/m and two identical shuttle masses with $m = 30$ μg. A force $f = 5\sin(2t)$ μN acts on the body on the right, as shown in the figure. In addition, an initial displacement $y_1(0) = -1.5 \cdot 10^{-11}$ m is applied to the body on the left of the figure. Consider that damping acts on both masses, with a damping coefficient $c = 0.1$ N s/m. Find a state-space model of the system and use MATLAB to plot its time response.

Solution

To use MATLAB for the time-domain solution, the problem is divided into two subproblems: it is considered first that the system is under the action of the force alone, then that only the initial conditions are applied. Since the system is linear, the two individual solutions are then added to obtain the total (actual) solution. The lumped-parameter model corresponding to the micromechanical system is shown in Figure 8.14, being formed of two masses m, three dampers c, and three springs k. The force f is also indicated in the figure.

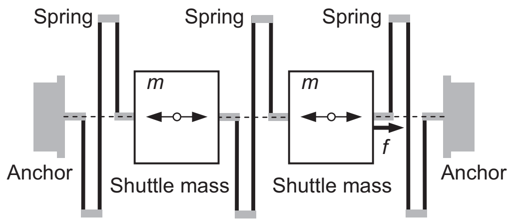

FIGURE 8.13

Two-Mass, Three-Spring Mechanical Microsystem.

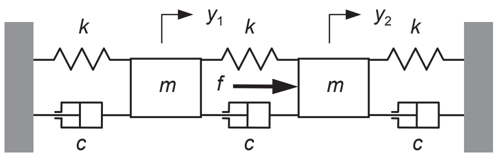

FIGURE 8.14

Lumped-Parameter Model of Two-DOF Translatory Mechanical System.

The following dynamic equations of this system can be derived using known methods, such as Newton's second law of motion or Lagrange's equations:

$$\begin{cases} m \cdot \ddot{y}_1 = -c \cdot \dot{y}_1 - k \cdot y_1 - c \cdot (\dot{y}_1 - \dot{y}_2) - k \cdot (y_1 - y_2) \\ m \cdot \ddot{y}_2 = -c \cdot \dot{y}_2 - k \cdot y_2 - c \cdot (\dot{y}_2 - \dot{y}_1) - k \cdot (y_2 - y_1) + f \end{cases}$$

$$\text{or} \quad \begin{cases} \ddot{y}_1 = -\dfrac{2k}{m} \cdot y_1 - \dfrac{2c}{m} \cdot \dot{y}_1 + \dfrac{k}{m} \cdot y_2 + \dfrac{c}{m} \cdot \dot{y}_2 \\[2ex] \ddot{y}_2 = \dfrac{k}{m} \cdot y_1 + \dfrac{c}{m} \cdot \dot{y}_1 - \dfrac{2k}{m} \cdot y_2 - \dfrac{2c}{m} \cdot \dot{y}_2 + \dfrac{1}{m} \cdot f \end{cases} \tag{8.113}$$

The following state variables are selected:

$$x_1 = y_1; \quad x_2 = \dot{y}_1; \quad x_3 = y_2; \quad x_4 = \dot{y}_2 \tag{8.114}$$

the input is the force, that is $u = f$, whereas the output is formed of the two displacements y_1 and y_2. Combination of Eqs. (8.113) and (8.114) generates the state equation:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2 \cdot \dfrac{k}{m} & -2 \cdot \dfrac{c}{m} & \dfrac{k}{m} & \dfrac{c}{m} \\ 0 & 0 & 0 & 1 \\ \dfrac{k}{m} & \dfrac{c}{m} & -2 \cdot \dfrac{k}{m} & -2 \cdot \dfrac{c}{m} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{m} \end{Bmatrix} \cdot u \tag{8.115}$$

which defines the $[A]$ and $[B]$ matrices. The output equation is determined from the first and third Eq. (8.114):

$$\begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \cdot u \tag{8.116}$$

which identifies the $[C]$ and $[D]$ matrices. Eq. (8.116) relates the initial values of the output and state vectors as follows: $\{y_1(0)\ y_2(0)\}^T = [C]\cdot\{x_1(0)\ x_2(0)\ x_3(0)\ x_4(0)\}^T$. Because $y_1(0) \neq 0$ and $y_2(0) = 0$, it follows that $x_1(0) = y_1(0) \neq 0$ and $x_3(0) = y_2(0) = 0$. As a consequence, the initial-condition vector can be selected as $\{x(0)\} = \{y_1(0)\ 0\ 0\ 0\}^T$.

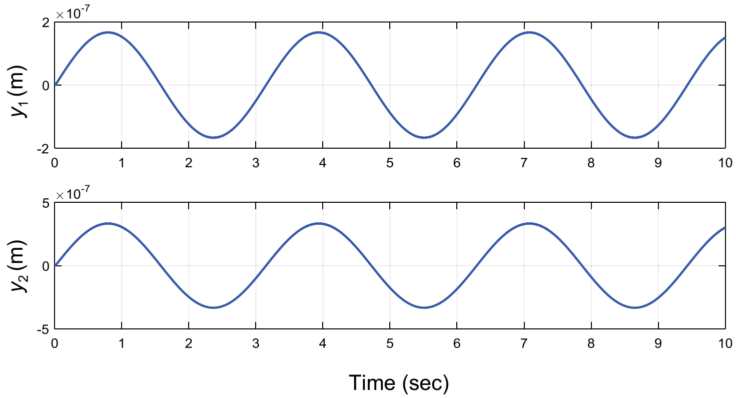

FIGURE 8.15

Plot of Displacement Outputs for the Two-DOF Mechanical System.

Figure 8.15 shows the response curves of the mechanical microsystem; the plots were obtained by the MATLAB code:

```
>> m = 30e-9; c = 0.1; k = 10;
>> t = 0:0.001:10;
>> u = 5e-6*sin(2*t);
>> A = [0,1,0,0;-2*k/m,-2*c/m, k/m, c/m;0,0,0,1;k/m, c/m,-2*k/m,-2*c/m];
>> B = [0;0;0;1/m]; C = [1,0,0,0;0,0,1,0]; D =[0;0];
>> sys = ss(A,B,C,D);
>> % individual response 'yf' to forcing
>> [yf,t,x] = lsim(sys,u,t);
>> % individual response 'yic' to initial conditions
>> y01 = -1.5e-11;
>> x0 = [y01;0;0;0];
>> [yic,t,x] = initial(sys,x0,t);
```

```
>> % total response 'y' as superposition of individual 'yf' and 'yic'
>> y = yf + yic;
>> subplot(2 1 1);
>> % plot of y_1 — first column of y
>> plot(t,y1(:,1))
>> subplot(2 1 2);
>> % plot of y_2 — second column of y
>> plot(t, y1(:,2))
```

8.3.3 Simulink Approach

Simulink offers an elegant environment for creating and simulating state-space models. The minimum configuration of a Simulink state-space model is formed of an input block, the state-space block, and an output (visualization/plot) block. Let us analyze a couple of examples of formulating and solving state-space problems by means of Simulink.

Example 8.17

A state-space model is defined by the matrices

$$A = \begin{bmatrix} -0.001 & -80 \\ 1 & 0 \end{bmatrix};\ B = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix};\ C = \{0 \quad 1\};\ D = \{0 \quad 0\}$$

The input to this system has two components: $u_1 = 20–18 \cdot e^{-0.5\,t}$, and u_2 is a pulse with an amplitude of 20, period of 2.5 s, and pulse width of 1.25 s. The initial conditions of the problem are $\{x(0)\} = \{1\ 2\}^T$. Use Simulink to plot the system response.

Solution

A new model window is opened and the following blocks are copied to it from the `Library` window: a `Ramp` block (with a `Slope` of 1 to model the time vector) and a `Pulse Generator` block (with an `Amplitude` of 20, `Period` of 2.5, and `Pulse Width` of 50%) from the `Sources` category. The `Fcn` (Function) block is taken from the `User-Defined Functions` library with the following `Expression: 20-18*exp(-0.5*u(1))`. The `State-Space` block is found in the `Continuous` category, and three `To Workspace` blocks are copied from the `Sinks` category. The system is a MISO (multiple-input/single-output) system, as the dimensions of the matrices indicate. We therefore need to use two signals combined into a single input, since the Simulink state-space operator works as a SISO one. The Simulink `Mathematical Operations` library possesses the `Vector Concatenate` operator that accepts several vectors (or matrices) as the input, which it combines into a single vector (or matrix) at the output, without modifying the original components. Configuration of the state-space block (which you have to double click) is done by filling in the following data.

```
Function Block Parameters: State-Space Window
A: [—0.001, —80; 1, 0]
B: [1, —1; 0, 0]
C: [0, 1]
D: [0, 0]
Initial conditions: [1; 2]
```

The Simulink block diagram is shown in Figure 8.16.

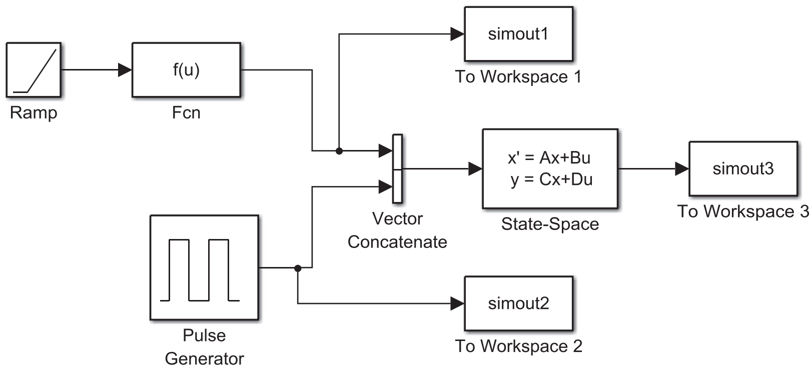

FIGURE 8.16

Simulink Block Diagram of the Two-Input, One-Output State-Space Model.

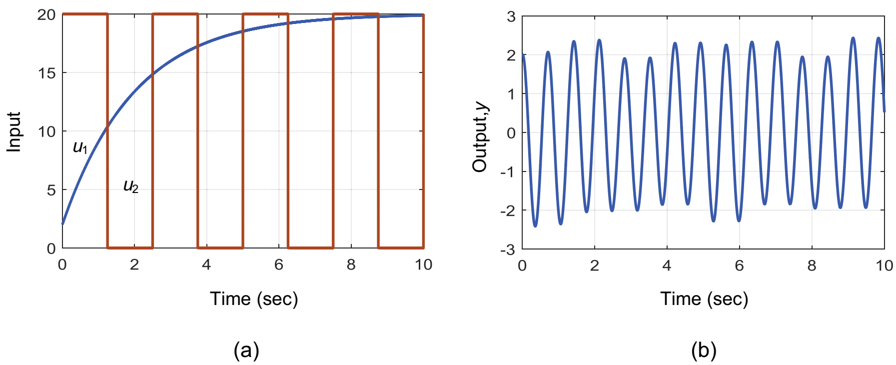

FIGURE 8.17

Simulink Plots of (a) The Pulse and Exponential Input Signals; (b) Output Signal.

Figure 8.17 contains the plots of the two inputs and the one of the output — all these signals were exported to MATLAB's workspace to be plotted as functions of time.

Note: An `LTI System` block (as discussed in Example 7.18 in the context of transfer function) can be used instead of the `State-Space` block and a `Mux` block instead of the `Vector Concatenate` block — the interested reader can test that variant. ■

Example 8.18

A body of mass $m = 1$ kg slides on a horizontal surface with friction. The body is connected to a spring of stiffness $k = 50$ N/m whose other end is fixed. The coefficient of static friction is $\mu_s = 0.3$, the coefficient of kinematic friction is $\mu_k = 0.2$, and the gravitational acceleration is $g = 9.81$ m/s^2. An initial displacement $y_0 = 0.9$ m is applied to the body. Find a state-space model of this mechanical system and plot its output (time-dependent displacement $y(t)$) using Simulink.

Solution

For the mass to vibrate, the initial displacement of 0.9 m should be larger than a threshold value determined from the static equilibrium between the spring force and the static friction force, which is $y_s = \mu_s \cdot m \cdot g/k = 0.0589$ m—this condition is definitely complied with. The free-body diagram of the body is shown in Figure 8.18(a) for the situation where the displacement and velocity have identical directions, whereas Figure 8.18(b) sketches the free-body diagram for the case where the velocity direction switches.

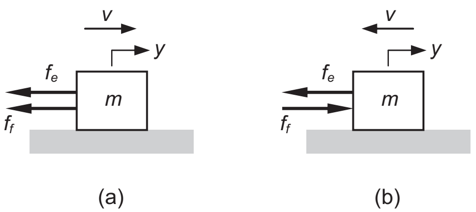

FIGURE 8.18

Free-Body Diagram for Body Under the Action of Elastic and Friction Forces for Velocity and Displacement Having (a) Identical Directions; (b) Opposite Directions.

Applying Newton's second law of motion for the two cases results in the following equations:

$$\begin{cases} m \cdot \ddot{y} = -k \cdot y - \mu_k \cdot m \cdot g & \text{for } y \cdot \dot{y} > 0 \\ m \cdot \ddot{y} = -k \cdot y + \mu_k \cdot m \cdot g & \text{for } y \cdot \dot{y} < 0 \end{cases} \quad \text{or} \quad \ddot{y} = -\frac{k}{m} \cdot y - \operatorname{sgn}(\dot{y}) \cdot \mu_k \cdot g \tag{8.117}$$

Eq. (8.117) took into account that the elastic (spring) force is $f_e = k \cdot y$ and the friction force is $f_f = \mu_k \cdot m \cdot g$. The new function *signum*, denoted as $\operatorname{sgn}(\dot{y})$ in Eq. (8.117), assumes a value of $+1$ when the velocity is positive (has the same direction as y) and a value of -1 when the velocity becomes negative (has a direction opposing the y direction). As a consequence, the last form of Eq. (8.117) captures both situations described by the two individual Eq. (8.117). Note also that Eq. (8.117) is nonlinear due to sgn function that is applied to the dependent variable y.Considering the friction action as the input, which is $u = \operatorname{sgn}(\dot{y}) \cdot \mu_k \cdot g$, and the state variables $x_1 = y;\ x_2 = \dot{y}$, the following state equation results via Eq. (8.117)

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & 0 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ -1 \end{Bmatrix} \cdot u \tag{8.118}$$

Because $y = x_1$, the output equation is

$$y = \{1 \quad 0\} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + 0 \cdot u \tag{8.119}$$

which also indicates that

$$y(0) = \{1 \quad 0\} \cdot \begin{Bmatrix} x_1(0) \\ x_2(0) \end{Bmatrix} = x_1(0) \tag{8.120}$$

The initial state variable vector is therefore as follows: $\{x(0)\} = \{x_1(0)\ x_2(0)\}^T = \{y(0)\ 0\}^T$.

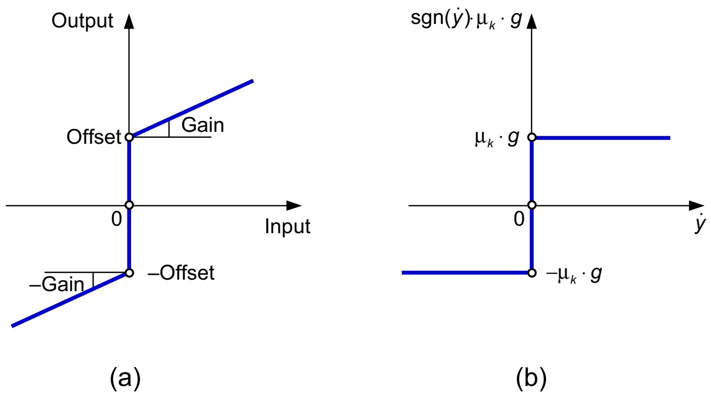

FIGURE 8.19

Simulink Nonlinear Coulomb and Viscous Friction: (a) Generic Structure; (b) Particular Structure.

MATLAB possesses the `Coulomb and Viscous Friction` block in the `Discontinuities` library. This block, which is shown in Figure 8.19(a), can model the sgn function in a more generic fashion by means of an offset and a gain. The input—output relationship of this Simulink block is expressed as follows:

$$\text{Output} = \text{sgn}(\text{Input}) \cdot (\text{Gain} \cdot |\text{Input}| + \text{Offset}) \tag{8.121}$$

In our particular application, the output from the block is $u = \text{sgn}(\dot{y}) \cdot \mu_k \cdot g$, which indicates that the gain is zero, whereas the offset is $\mu_k \cdot g$, as shown in Figure 8.19(b).

Figure 8.20 shows the block diagram of the state-space model described by Eqs. (8.118) and (8.119). The `State-Space` block has the matrices `A`, `B`, `C`, and `D` that were defined in the same Eqs. (8.118) and (8.119). In addition, the `Initial Conditions` of this block are [0.9; 0]. In the `Coulomb and Viscous Friction` block, the `Coulomb friction value (offset)` is $\mu_k \cdot g = 1.962$, whereas the `Gain` is 0. Note that the input to this block is the velocity, which results from the `Derivative` block taken from the `Continuous` library. The output y is exported to MATLAB, where the plot $y(t)$ illustrated in Figure 8.21 is obtained. As it is noticed, the body motion stops at a distance from the equilibrium (zero) position, and this offset has the magnitude

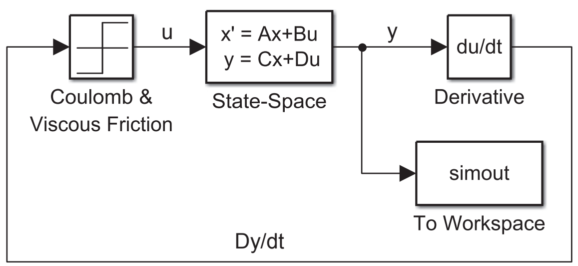

FIGURE 8.20

Simulink Block Diagram of the State-Space Model Without and With Coulomb-Friction Nonlinear Effect.

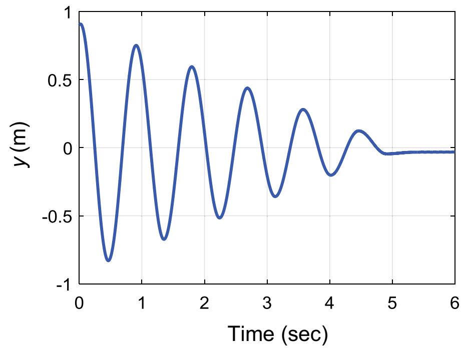

FIGURE 8.21

Simulink Plot of State-Space Model Time Response for Nonlinear Mechanical System With Coulomb Friction.

of the static equilibrium spring deformation, which was calculated at the beginning of this example as $y_s = 0.0589$ m.

SUMMARY

This chapter introduces the concept of state-space as an approach to model and determine the response of dynamic systems in the time domain. Using a matrix formulation, the state-space procedure is an alternate method of characterizing mostly MIMO systems defined by a large number of coordinates (DOF). Different state-space algorithms are applied, depending on whether the input has time derivatives or no time derivatives. Methods of calculating the free response with nonzero initial conditions and the forced response using the state-space approach are presented as well. Conversion between state-space and transfer function or zero-pole-gain models is also studied, both analytically and by means of MATLAB-specialized commands. The chapter also discusses the principles of linearizing nonlinear state-space models. The material includes the application of specialized MATLAB commands to solve state-space formulated problems and examples of using Simulink to model and solve linear/nonlinear system dynamics problems by the state-space approach.

PROBLEMS

8.1 Find a state-space model for the translatory mechanical system of Figure 8.22, considering that the input is the displacement u of the chain's free end and the output is the displacement y. Derive another state-space model when the output is formed of y and of the velocity dy/dt.

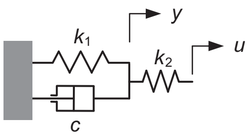

FIGURE 8.22

Translatory Mechanical System With Springs, Damper, and Displacement Input.

8.2 A single-mesh electrical system comprises an inductor L, a capacitor C, and a source voltage v. Derive a state-space model where the input is v, the output is the current i, and the state variables are the voltage across the capacitor v_C and the current i.

8.3 The electric system of Figure 8.23 comprises the resistor R and the capacitor C. Find a state-space model for this system by starting from the time-domain model of this system and using one state variable only.

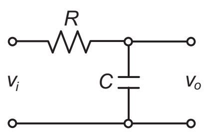

FIGURE 8.23

Single-Stage Capacitive—Resistive Electrical System.

8.4 Find a state-space representation for the electrical circuit shown in Figure 8.24 using only one state-space variable. Known are the resistors R_1, R_2, R_3 and the capacitor C. The input is the source current i, and the output is the voltage difference across any of the four electrical elements.

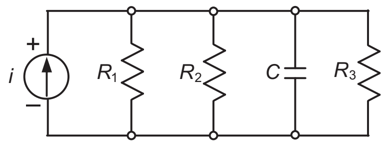

FIGURE 8.24

Electrical Circuit With Current Source.

8.5 Starting from the time-domain model of the operational-amplifier electrical circuit shown in Figure 8.25, which contains the resistor R and the inductor L, derive a state-space model using one state variable when v_i is the input and v_o is the output.

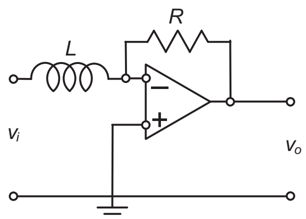

FIGURE 8.25

Operational-Amplifier Electrical System With Resistor and Inductor.

8.6 The pneumatic system of Figure 8.26 consists of a pneumatic resistance R_g and a gas container whose capacity is C_g. The input (container) pressure is p_i, and the output pressure is p_o. Derive a state-space model for this system.

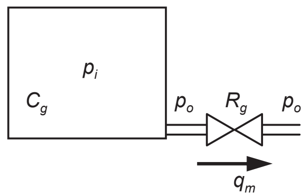

FIGURE 8.26

Pneumatic System With Resistance and Capacitance.

8.7 Derive a state-space model for the one-room thermal system of Figure 8.27. Consider that the input is formed of the outside temperature θ_o and the source heat flow rate q_i, whereas the output is the inside temperature θ_i. The enclosed space has a thermal capacity C_{th}, and each of the four identical walls has a thermal resistance R_{th}.

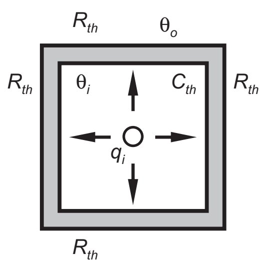

FIGURE 8.27

Four-Wall, One-Room Thermal System.

8.8 Formulate a state-space model for the liquid system of Figure 8.28, which comprises a tank of capacitances C_l and two valves of resistances R_{l1} and R_{l2}. The input to the system is the pressure p_i, and the output is the pressure p at the bottom of the tank.

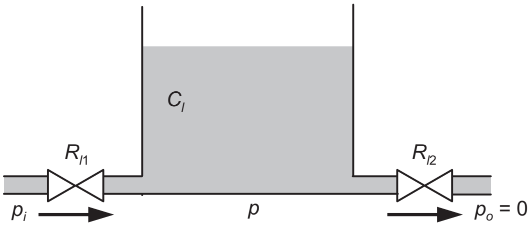

FIGURE 8.28

Liquid-Level System With One Tank and Two Valves.

8.9 The MEMS of Figure 8.29 is formed of two shuttle masses m_1 and m_2 coupled by a serpentine spring of stiffness k and supported separately by two pairs of identical beam springs—each beam has a stiffness k_1. The shuttle masses are subjected to viscous damping individually through substrate interaction—the damping coefficient is c, and an electrostatic force f acts on m_1. Use a lumped-parameter model of this MEMS device and obtain a state-space model for it by considering that the input is f, and the output vector comprises the two masses' displacements and velocities.

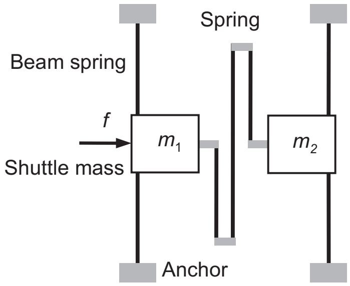

FIGURE 8.29

Translatory-Motion MEMS Device.

8.10 Find a state-space model for the translatory mechanical system of Figure 8.30 considering that the input is the force f applied to the mass m and, the output vector contains all relevant displacements. Known are also the spring stiffnesses k_1, k_2, and the damper's viscous damping coefficient c.

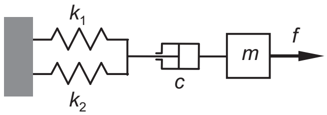

FIGURE 8.30

Translatory Mechanical System With Springs, Damper, Mass, and Force Input.

8.11 Derive a state-space model for the electrical system of Figure 8.31, where the input is the voltage v_i, and the output is the voltage v_o. Consider known the resistance R, inductance L, and capacitance C.

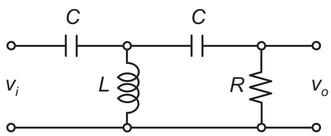

FIGURE 8.31

Two-Stage Electrical System.

8.12 The liquid system shown in Figure 8.32 is formed of two tanks of capacitances C_{l1} and C_{l2} and two valves of resistances R_{l1} and R_{l2}. Formulate a state-space model for it by considering the input to the system is the volume flow rate q_i and the pressure p_3; the output consists of the pressures at the tanks' bottoms, p_1 and p_2.

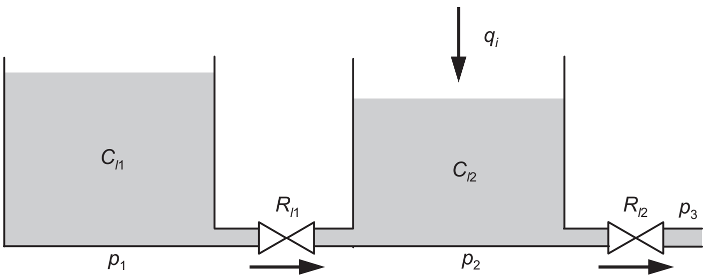

FIGURE 8.32

Two-Tank Liquid-Level System.

8.13 Find state-space models for the SISO dynamic systems represented by the following differential equations, where y is the output, and u is the input:

(i) $\dddot{y} + 5\ddot{y} + 3\dot{y} + 2y = 3\dot{u}$

(ii) $2\ddot{y} - \dot{y} + 4y = 6\ddot{u}$

(iii) $\dddot{y} + \ddot{y} - 7\dot{y} + 3y = \ddot{u} + 2\dot{u} + 4u$

8.14 Derive a state-space model for the rotary mechanical system of Figure 8.33, which is formed of a cylinder with mass moment of inertia J, a spring of stiffness k, and two dampers defined by the damping coefficients c_1 and c_2. The input is the rotation angle θ_u, and the output is the cylinder rotation angle.

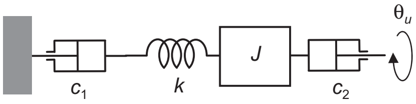

FIGURE 8.33

Rotary Mechanical System.

8.15 Obtain a state-space model for the translatory mechanical system of Figure 8.34 assuming the input is the displacement u, and the output is the displacement y. Known are the mass m, the damper viscous damping coefficient c, and the spring stiffnesses k_1 and k_2.

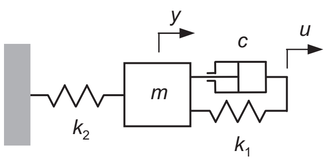

FIGURE 8.34

Translatory Mechanical System With Displacement Input.

8.16 Obtain an state-space model for the electrical system of Figure 8.35, where the applied voltage v_i is the input, and the voltage v_o is the output. Known are the resistances R_1, R_2, and inductance L.

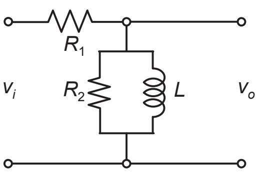

FIGURE 8.35

One-Stage Electrical System With Resistors and Inductor.

8.17 Consider the operational-amplifier system shown in Figure 8.36. Knowing the resistances R_1, R_2, the inductance L, and the capacitance C:

(i) Find a state-space model for this circuit for the voltage v_i being the input, and the voltage v_o being the output

(ii) Convert the state-space model into the corresponding transfer function.

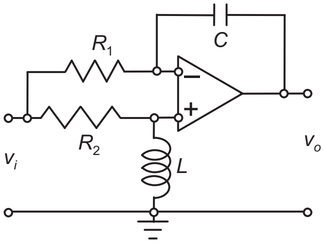

FIGURE 8.36

Operational-Amplifier Electrical System.

8.18 Convert the transfer function of the electrical system sketched in Figure 7.2 of Example 7.1 into a state-space model using the analytical approach.

8.19 Same question as in Problem 8.18 for the mechanical system of Example 7.9 and shown in Figure 7.14.

8.20 Convert the zero-pole-gain model of Example 7.4 into a state-space model using both the analytical approach and MATLAB.

8.21 A unity-gain dynamic system has the following zeroes: 0 (simple) and -1 (double), and the poles: $-2+j$ (double) and $-4-j$ (simple). Determine a state-space model corresponding to the given zero-pole-gain model by using the analytical approach. Use MATLAB to confirm that the state-space model can be converted back into the original zero-pole-gain model.

8.22 Use analytical calculation to obtain a state-space model from the transfer function: $G(s)=\frac{2s+1}{s^3+s+1}$.

8.23 Transform the state-space model of the electrical system of Problem 8.16 and pictured in Figure 8.35 into a transfer function for $R_1 = 250\,\Omega$, $R_2 = 220\,\Omega$, and $L = 0.5$ H.

8.24 Convert the transfer function matrix

$$[G(s)] = \begin{bmatrix} \dfrac{s}{s^2+s+1} & \dfrac{s+1}{s^2+s+1} & \dfrac{2s+3}{s^2+s+1} \\ \dfrac{1}{s^2+s+1} & \dfrac{2}{s^2+s+1} & \dfrac{4s}{s^2+s+1} \end{bmatrix}$$

into a state-space model by using MATLAB.

8.25 Convert the state-space model of the mechanical system illustrated in Figure 8.2 of Example 8.3 into a transfer function model using the analytical transformation equation.

8.26 A state-space model is defined by the following matrices:

$$[A] = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}; \quad [B] = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}; \quad [C] = \{3 \quad 1\}; \quad [D] = \{1 \quad 0\}$$

Convert this model into a transfer function model using analytical calculation and also applying MATLAB conversion.

8.27 Same question as in Problem 8.26 for the system defined by the matrices:

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad [B] = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}; \quad [C] = \{1 \quad 0 \quad 3\}; \quad [D] = 0$$

8.28 The transfer function of a SISO dynamic system is $G(s) = (s + 6)/(s^2 + 10s + 81)$. Use MATLAB to determine the corresponding zero-pole-gain model, and then utilize the resulting zero-pole-gain model to derive a state-space model. Also obtain a state-space model directly from the transfer function model. If the two state-space models are different, explain the discrepancy.

8.29 A point mass m and a nonlinear spring are attached to the lever sketched in Figure 8.37, which vibrates in a horizontal plane. Assume the elastic (spring) force is defined as $f_e = 0.9\,y + 0.1 \cdot y^3$ (where y represents the displacement of the spring end attached to and perpendicular to the lever) and the rod is massless. For small rotations of the rod around its pin axis, obtain a linearized state-space model by considering small vibrations about the equilibrium position.

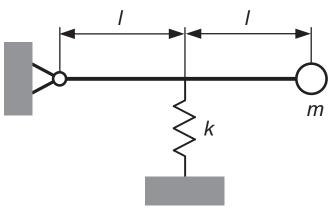

FIGURE 8.37

Mass-Spring Lever System.

8.30 Solve Problem 8.29 when the system vibrates in a vertical plane and the lever has a mass m_l. Consider small motions around the static-equilibrium position of the lever and take into consideration the gravitational acceleration g.

8.31 Use a lumped-parameter model for the single-DOF mechanical micro-accelerometer of Figure 8.38. Find a transfer function model of this system and convert it into a state-space model. The five identical massless beams have

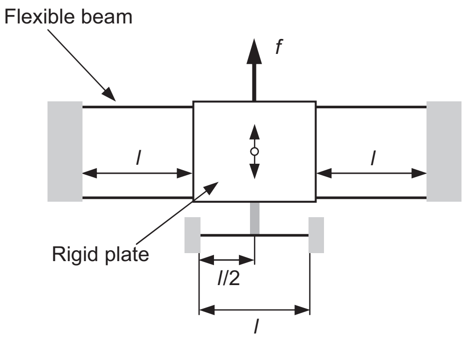

FIGURE 8.38

Microaccelerometer With Plate and Beam Springs.

a length $l = 30$ μm, their cross-section is square with a side of $a = 1.2$ μm, Young's modulus is $E = 160$ GPa, and the mass of the central plate is 180 μg. Use the state-space model and MATLAB to calculate the mass displacement when a force $f = 2 \cdot 10^{-8}\delta(t)$ N (where $\delta(t)$ is the unit impulse) is applied to the plate. Ignore damping and inertia contributions from the beams.

8.32 Use complex impedances to determine the transfer function of the electrical system shown in Figure 8.39, and then convert the transfer function model

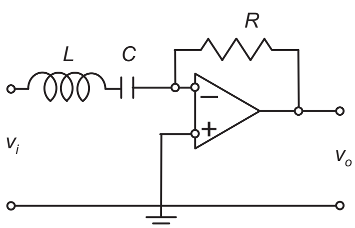

FIGURE 8.39

Single-Stage Operational-Amplifier Electrical System.

into a state-space model by analytical calculation. Use MATLAB to plot the output voltage as a function of time with the aid of the state-space model. Known are $R = 360\ \Omega$, $L = 0.6$ H, $C = 50$ μF, and $v_i = 100 \cdot \sin(12t)$ V.

8.33 **(i)** Determine the transfer function of the electrical system of Figure 8.40 for an input voltage v_i and an output voltage v_o.

(ii) Convert the transfer function into a state-space model. Use the state-space model to plot v_o for $R_1 = 120\ \Omega$, $R_2 = 160\ \Omega$, $C_1 = 200$ μF, $C_2 = 300$ μF, and $v_i = 50 \cdot \left(1 + 10e^{-t}\right)$ V.

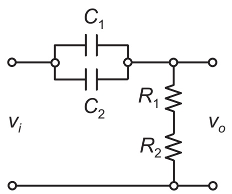

FIGURE 8.40

Resistive—Capacitive Electrical System.

8.34 Consider an initial rotation angle of 6 degrees is applied to the pivoting rod of the lever system studied in Problem 8.29 and shown in Figure 8.37. Solve the linearized state-space model using the state-transition matrix method; also use Simulink to solve both the original nonlinear and the linearized state-space models. Known is $m = 0.7$ kg.

8.35 Apply the state-space approach and the state-transition matrix to find the currents in the circuit of Figure 8.41 when an initial charge $q_0 = 0.5\ C$ is applied to the capacitor. Plot these currents in terms of time when known are $R_1 = 150\ \Omega$, $R_2 = 180\ \Omega$, $L = 0.6$ H, and $C = 160\ \mu$F.

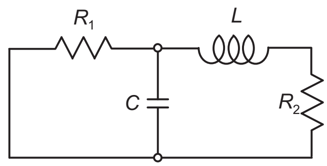

FIGURE 8.41

Two-Mesh Electrical System With Initial Charge on Capacitor.

8.36 The homogeneous state-space model of a dynamic system is defined by the matrices

$$[A] = \begin{bmatrix} 0 & 1 \\ -110 & -0.3 \end{bmatrix}; \quad [C] = \{2 \quad 0\}$$

Use MATLAB to plot the system response $y(t)$ if the following initial condition is applied: $y(0) = 0.1$.

8.37 A homogeneous state-space model has the nonzero matrices:

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad [C] = \{1 \quad -1 \quad 2\}$$

Using the state-transition method, determine the system's response $y(t)$ and plot it with respect to time for the initial condition: $\{x(0)\} = \{2 \quad -1 \quad 1\}^T$.

8.38 Use the state-transition matrix method to express and plot the output of the dynamic system described by the state-space model of Example 8.10. Consider zero initial conditions and an input is $u(t) = 10 \cdot e^{-2t} \cdot \sin(15t)$.

8.39 Consider the two-room thermal system of Figure 8.42. Derive a state-space model for the system and use MATLAB to plot the two room temperatures θ_1 and θ_2 when the outside temperature is $\theta_o = 36°C$ and the initial room temperatures are $\theta_1(0) = 28°C$ and $\theta_2(0) = 15°C$. All walls are identical and of dimensions 9 m · 5 m · 0.2 m; the wall material thermal conductivity is $k = 0.055$ W/(m deg). The rooms' thermal capacitance is $C_{th} = 130{,}000$ J/deg.

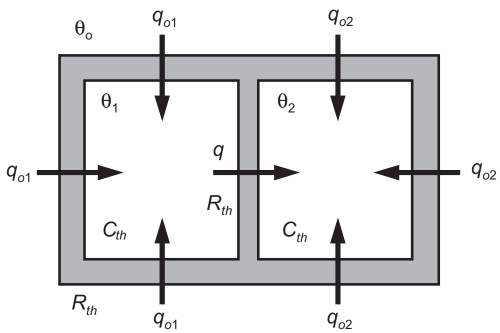

FIGURE 8.42

Two-Room Thermal System.

8.40 Find a state-space model for the pneumatic system of Figure 8.43 knowing the input pressures $p_{i,1} = 40$ atm and $p_{i,2} = 30$ atm. Known also are the pneumatic resistances $R_{g1} = 2100\ \mathrm{m}^{-1}\ \mathrm{s}^{-1}$, $R_{g2} = 3200\ \mathrm{m}^{-1}\ \mathrm{s}^{-1}$, $R_{g3} = 1500\ \mathrm{m}^{-1}\ \mathrm{s}^{-1}$, and tank capacitances $C_{g1} = 0.0006\ \mathrm{m\ s}^2$, $C_{g2} = 0.0005\ \mathrm{m\ s}^2$. Use the state-transition matrix method, as well as Simulink, to find and to plot the pressures in the two containers.

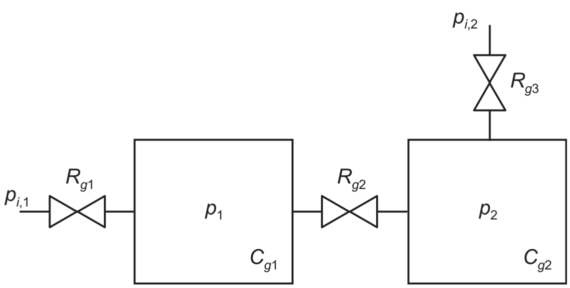

FIGURE 8.43

Pneumatic System With Two Containers and Ductwork.

8.41 A dynamic system is defined by the state-space matrices

$$[A] = \begin{bmatrix} 1 & 0 \\ -4 & -65 \end{bmatrix}; \ [B] = \begin{Bmatrix} -2 \\ 0 \end{Bmatrix}; \ [C] = \{0 \quad 1\}; \ [D] = 3.$$

Find the system response and plot it against time when an input $u = 4 \cdot \sin(30t)$ is applied to the system with zero initial conditions; use both MATLAB and Simulink for that.

8.42 A parallel *RLC* circuit is connected to a voltage source that provides a sinusoidal voltage $v = 90 \cdot \sin(16t)$ V. It is also known that $L = 0.3$ H and $C = 400$ μF. Plot the currents in the three components by using Simulink and its state-space capability, when an initial charge $q_0 = 0.2$ C is applied to the capacitor and

(i) The resistor is linear with $R = 160\ \Omega$.

(ii) The resistor defined at (i) has a saturation nonlinearity (discontinuity) defined by the limit voltages of -60 and 70 V.

8.43 Two bodies of masses $m_1 = 0.8$ kg and $m_2 = 0.5$ kg are connected by a spring and a damper connected in parallel defined by $c = 60$ N s/m and $k = 140$ N/m. The bodies slide on a horizontal surface with friction. The coefficient of static friction is $\mu_s = 0.3$, and the coefficient of kinematic friction is $\mu_k = 0.2$. A sinusoidal force $f = 50 \cdot \sin(12t)$ N acts on the body of mass m_2. Find a state-space model of this mechanical system and determine its time response using Simulink when nonlinear effects of Coulomb friction are considered.

Suggested Reading

H. Klee and R. Allen, *Simulation of Dynamic Systems with MATLAB and Simulink*, 2nd Ed. CRC Press, Boca Raton, FL, 2011.

N. S. Nise, *Control System Engineering*, 7th Ed. John Wiley & Sons, New York, 2015.

B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, 10th Ed. John Wiley & Sons, New York, 2017.

P. Marchand and O. T. Holland, *Graphics and GUIs with MATLAB®*, 3rd Ed. Chapman & Hall/CRC, Boca Raton, FL, 2003.

K. Ogata, *System Dynamics*, 4th Ed. Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

D. G. Luenberger, *Introduction to Dynamics Systems*, John Wiley & Sons, New York, 1979.