Chapter Three

State Space Approach

The state space approach has been introduced in Section 1.3. Due to its fundamental importance for control systems, the state space technique will be considered thoroughly in this chapter. Both continuous- and discrete-time linear time invariant systems will be presented. It has already been pointed out that the state space technique represents the modern approach to control system theory and its applications. The state space approach is very convenient for representation of high-order dimensional and complex systems, and extremely efficient for numerical calculations since many efficient and reliable numerical algorithms developed in mathematics, especially within the area of numerical linear algebra, can be used directly. In addition, the state space form is the basis for introducing system controllability and observability concepts and many modern control theory techniques.

The state space model of a continuous-time linear system is represented by a system of n linear differential equations. In matrix form, it is given by

$$\frac{d}{dt}\mathbf{x}(t) = \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_{\text{o}} \tag{3.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{3.2}$$

where $\mathbf{x} \in \Re^n$, $\mathbf{u} \in \Re^r$, and $\mathbf{y} \in \Re^p$ are, respectively, vectors of system states, control inputs, and system outputs. The matrix $\mathbf{A}^{n \times n}$ describes the *internal* behavior of the system, while matrices $\mathbf{B}^{n \times r}$, $\mathbf{C}^{p \times n}$, and $\mathbf{D}^{p \times r}$ represent connections between the *external world* and the system. If there are no direct paths between inputs and outputs, which is often the case, the matrix $\mathbf{D}^{p \times m}$ is zero. It is assumed in this book that all matrices in (3.1) and (3.2) are time invariant. Studying linear control systems with time varying coefficient matrices requires knowledge of some advanced topics in mathematics (see for example Chen, 1984; see also Section 10.1).

The state space model for linear discrete-time control systems has exactly the same form as (3.1) and (3.2) with differential equations replaced by difference equations, that is

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k), \qquad \mathbf{x}(0) = \mathbf{x}_o \tag{3.3}$$

$$\mathbf{y}(k) = \mathbf{C}_d\mathbf{x}(k) + \mathbf{D}_d\mathbf{u}(k) \tag{3.4}$$

All vectors and matrices defined in (3.3) and (3.4) have the same dimensions as corresponding ones given in (3.1) and (3.2). In this chapter, we present and derive in detail the main state space concepts for continuous-time linear control systems and then give the corresponding interpretations in the discrete-time domain.

The chapter is organized as follows. In Section 3.1 several systematic methods for obtaining the state space form from differential equations and transfer functions are developed. The time response of linear systems given in the state form is considered in Section 3.2. The corresponding results for discrete-time systems, and the procedure for discretization of continuous-time systems leading to discrete-time models, are given in Section 3.3. The concepts of the system characteristic equation, eigenvalues, and eigenvectors and their use in control system theory are presented in Section 3.4. At the end of the chapter, in Section 3.5, three MATLAB laboratory experiments are outlined.

Chapter Objectives

The dynamical systems considered in this book are either described by differential/difference equations or given in the form of system transfer functions. One of the goals is to present procedures for obtaining the state space forms either from differential/difference equations or from transfer functions. In that respect students will be exposed to four standard state space forms, known as canonical forms: the phase variable form or controller form, the observer form, the modal form, and the Jordan form.

Another important objective is to show students how to analyze linear systems given in the state space form, i.e. how to find responses (state variables and outputs) of the corresponding state space models to any input signal (control input). A working knowledge of undergraduate linear algebra and the basic theory of differential equations is helpful for complete understanding of this chapter. Some useful results on linear algebra are given in Appendix C. Students without a strong background in these topics may consult any undergraduate text, (for example, Fraleigh and Beauregard, 1990; Boyce and DiPrima, 1992).

3.1 State Space Models

In this section we study state space models of continuous-time linear systems. The corresponding results for discrete-time systems, obtained via duality with the continuous-time models, are given in Section 3.3.

The state space model of a continuous-time dynamic system can be derived either from the system model given in the time domain by a differential equation or from its transfer function representation. Both cases will be considered in this section. Four state space forms—the phase variable form (controller form), the observer form, the modal form, and the Jordan form—which are often used in modern control theory and practice, are presented.

3.1.1 The State Space Model and Differential Equations

Consider a general nth-order model of a dynamic system represented by an nth-order differential equation

$$\begin{aligned}\frac{d^n y(t)}{dt^n} &+ a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \cdots + a_1\frac{dy(t)}{dt} + a_0 y(t) \\ &= b_n\frac{d^n u(t)}{dt^n} + b_{n-1}\frac{d^{n-1}u(t)}{dt^{n-1}} + \cdots + b_1\frac{du(t)}{dt} + b_0 u(t)\end{aligned} \tag{3.5}$$

At this point we assume that all initial conditions for the above differential equation, i.e. $y(0^-), dy(0^-)/dt, ..., d^{n-1}y(0^-)/dt^{n-1}$, are equal to zero. We will show later how to take into account the effect of initial conditions.

In order to derive a systematic procedure that transforms a differential equation of order n to a state space form representing a system of n first-order differential equations, we first start with a simplified version of (3.5), namely we study the case when no derivatives with respect to the input are present

$$\frac{d^n y(t)}{dt^n} + a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \cdots + a_1\frac{dy(t)}{dt} + a_0 y(t) = u(t) \tag{3.6}$$

Introduce the following (easy to remember) change of variables

$$
\begin{aligned}
x_1(t) &= y(t) \\
x_2(t) &= \frac{dy(t)}{dt} \\
x_3(t) &= \frac{d^2y(t)}{dt^2} \\
&\vdots \\
x_n(t) &= \frac{d^{n-1}y(t)}{dt^{n-1}}
\end{aligned}
\tag{3.7}
$$

which after taking derivatives leads to

$$
\begin{aligned}
\frac{dx_1(t)}{dt} &= \dot{x}_1 = \frac{dy(t)}{dt} = x_2(t) \\
\frac{dx_2(t)}{dt} &= \dot{x}_2 = \frac{d^2y(t)}{dt^2} = x_3(t) \\
\frac{dx_3(t)}{dt} &= \dot{x}_3 = \frac{d^3y(t)}{dt^3} = x_4(t) \\
&\vdots \\
\frac{dx_n(t)}{dt} &= \dot{x}_n = \frac{d^ny(t)}{dt^n} \\
&= -a_0y(t) - a_1\frac{dy(t)}{dt} - a_2\frac{d^2y(t)}{dt^2} - \cdots - a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + u(t) \\
&= -a_0x_1(t) - a_1x_2(t) - \cdots - a_2x_3(t) - \cdots - a_{n-1}x_n(t) + u(t)
\end{aligned}
\tag{3.8}
$$

The state space form of (3.8) is given by

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix}
=
\begin{bmatrix}
0 & 1 & 0 & \cdots & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \cdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & \cdots & 0 & 1 \\
-a_0 & -a_1 & -a_2 & \cdots & \cdots & -a_{n-1}
\end{bmatrix}
\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix}
+
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)
\tag{3.9}
$$

with the corresponding output equation obtained from (3.7) as

$$y(t) = [1 \quad 0 \quad \cdots \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} \tag{3.10}$$

The state space form (3.9) and (3.10) is known in the literature as the *phase variable canonical form*.

In order to extend this technique to the general case defined by (3.5), which includes derivatives with respect to the input, we form an auxiliary differential equation of (3.5) having the form of (3.6) as

$$\frac{d^n\xi(t)}{dt^n} + a_{n-1}\frac{d^{n-1}\xi(t)}{dt^{n-1}} + \cdots + a_1\frac{d\xi(t)}{dt} + a_0\xi(t) = u(t) \tag{3.11}$$

for which the change of variables (3.7) is applicable

$$\begin{aligned} x_1(t) &= \xi(t) \\ x_2(t) &= \frac{d\xi(t)}{dt} \\ x_3(t) &= \frac{d^2\xi(t)}{dt^2} \\ &\vdots \\ x_n(t) &= \frac{d^{n-1}\xi(t)}{dt^{n-1}} \end{aligned} \tag{3.12}$$

and then apply the superposition principle to (3.5) and (3.11). Since $\xi(t)$ is the response of (3.11), then by the superposition property the response of (3.5) is given by

$$y(t) = b_0\xi(t) + b_1\frac{d\xi(t)}{dt} + b_2\frac{d^2\xi(t)}{dt^2} + \cdots + b_n\frac{d^n\xi(t)}{dt^n} \tag{3.13}$$

Equations (3.12) produce the state space equations in the form already given by (3.9). The output equation can be obtained by eliminating $d^n\xi(t)/dt^n$ from (3.13), by using (3.11), that is

$$\frac{d^n\xi(t)}{dt^n} = u(t) - a_{n-1}x_n(t) - \cdots - a_1x_2(t) - a_0x_1(t)$$

This leads to the output equation

$$y(t) = \left[(b_0 - a_0 b_n) \quad (b_1 - a_1 b_n) \quad \cdots \quad (b_{n-1} - a_{n-1} b_n) \right] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + b_n u(t) \tag{3.14}$$

It is interesting to point out that for $b_n = 0$, which is almost always the case, the output equation also has an easy-to-remember form given by

$$y(t) = \left[b_0 \quad b_1 \quad \cdots \quad b_{n-1} \right] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{3.15}$$

Thus, in summary, for a given dynamic system modeled by differential equation (3.5), one is able to write immediately its state space form, given by (3.9) and (3.15), just by identifying coefficients a_i and b_i, $i = 0, 1, 2, ..., n-1$, and using them to form the corresponding entries in matrices $\mathbf{A}$ and $\mathbf{C}$.

Example 3.1: Consider a dynamical system represented by the following differential equation

$$y^{(6)} + 6y^{(5)} - 2y^{(4)} + y^{(2)} - 5y^{(1)} + 3y = 7u^{(3)} + u^{(1)} + 4u$$

where $y^{(i)}$ stands for the ith derivative, i.e. $y^{(i)} = d^i y / dt^i$. According to (3.9) and (3.14), the state space model of the above system is described by the following matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -3 & 5 & -1 & 0 & 2 & -6 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = \left[4 \quad 1 \quad 0 \quad 7 \quad 0 \quad 0 \right], \quad \mathbf{D} = 0$$

◇

3.1.2 State Space Variables from Transfer Functions

In this section, we present two methods, known as direct and parallel programming techniques, which can be used for obtaining state space models from system transfer functions. For simplicity, like in the previous subsection, we consider only single-input single-output systems.

The resulting state space models may or may not contain all the modes of the original transfer function, where by transfer function modes we mean poles of the original transfer function (before zero-pole cancellation, if any, takes place). If some zeros and poles in the transfer function are cancelled, then the resulting state space model will be of reduced order and the corresponding modes will not appear in the state space model. This problem of system reducibility will be addressed in detail in Chapter 5 after we have introduced the system controllability and observability concepts.

In the following, we first use direct programming techniques to derive the state space forms known as the controller canonical form and the observer canonical form; then, by the method of parallel programing, the state space forms known as modal canonical form and Jordan canonical form are obtained.

The Direct Programming Technique and Controller Canonical Form

This technique is convenient in the case when the plant transfer function is given in a nonfactorized polynomial form

$$\frac{Y(s)}{U(s)} = \frac{b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{3.16}$$

For this system an auxiliary variable $V(s)$ is introduced such that the transfer function is split as

$$\frac{V(s)}{U(s)} = \frac{1}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{3.17a}$$

$$\frac{Y(s)}{V(s)} = b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0 \tag{3.17b}$$

The block diagram for this decomposition is given in Figure 3.1.

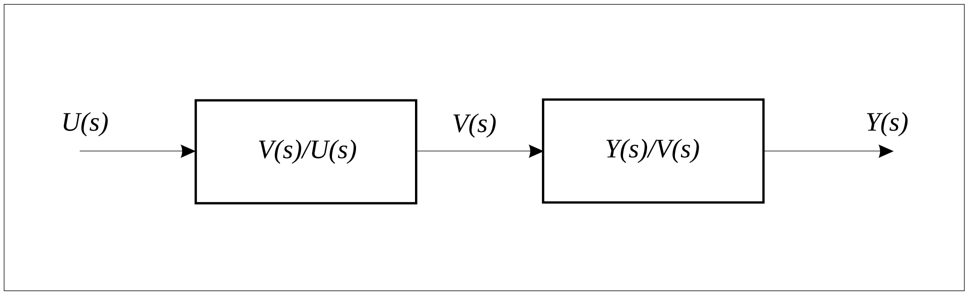

Figure 3.1: Block diagram representation for (3.17)

Equation (3.17a) has the same structure as (3.6), after the Laplace transformation is applied, which directly produces the state space system equation identical to (3.9). It remains to find matrices for the output equation (3.2). Equation (3.17b) can be rewritten as

$$Y(s) = b_n s^n V(s) + b_{n-1} s^{n-1} V(s) + \cdots + b_1 s V(s) + b_0 V(s) \tag{3.18}$$

indicating that $y(t)$ is just a superposition of $v(t)$ and its derivatives. Note that (3.17) may be considered as a differential equation in the operator form for zero initial conditions, where $s \equiv d/dt$. In that case, $V(s)$, $Y(s)$, and $U(s)$ are simply replaced with $v(t)$, $y(t)$, and $u(t)$, respectively.

The common procedure for obtaining state space models from transfer functions is performed with help of the so-called transfer function *simulation diagrams*. In the case of continuous-time systems, the simulation diagrams are elementary analog computers that solve differential equations describing systems dynamics. They are composed of integrators, adders, subtracters, and multipliers, which are physically realized by using operational amplifiers. In addition, function generators are used to generate input signals. The number of integrators in a simulation diagram is equal to the order of the differential equation under consideration. It is relatively easy to draw (design) a simulation diagram. There are many ways to draw a simulation diagram for a given dynamic system, and there are also many ways to obtain the state space form from the given simulation diagram.

The simulation diagram for the system (3.17) can be obtained by direct programming technique as follows. Take n integrators in cascade and denote their inputs, respectively, by $v^{(n)}(t), v^{(n-1)}(t), ..., v^{(1)}(t), v(t)$. Use formula (3.18) to construct $y(t)$, i.e. multiply the corresponding inputs $v^{(i)}(t)$ to integrators by the corresponding coefficients b_i and add them using an adder (see the top half of Figure 3.2, where $1/s$ represents the integrator block). From (3.17a) we have that

$$v^{(n)}(t) = u(t) - a_{n-1}v^{(n-1)}(t) - \cdots - a_1 v^{(1)}(t) - a_0 v(t)$$

which can be physically realized by using the corresponding feedback loops in the simulation diagram and adding them as shown in the bottom half of Figure 3.2.

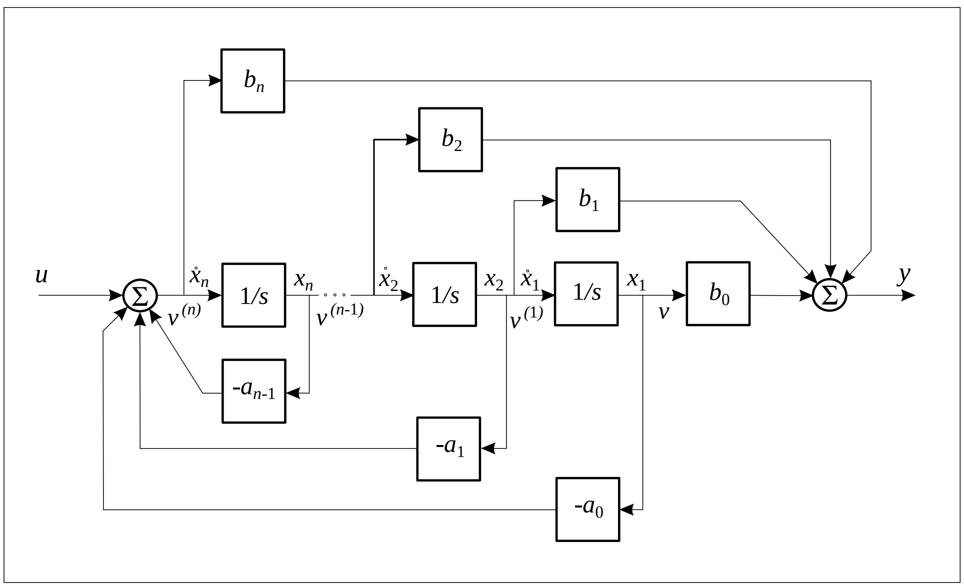

Figure 3.2: Simulation diagram for the direct programming technique (controller canonical form)

A systematic procedure to obtain the state space form from a simulation diagram is *to choose the outputs of integrators as state variables*. Using this convention, the state space model for the simulation diagram presented in Figure 3.2 is obtained in a straightforward way by reading and recording information from the simulation diagram, which leads to

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) \tag{3.19}$$

and

$$y(t) = [(b_0 - a_0 b_n) \quad (b_1 - a_1 b_n) \quad \cdots \quad (b_{n-1} - a_{n-1} b_n)]\mathbf{x}(t) + b_n u(t) \tag{3.20}$$

This form of the system model is called the *controller canonical form*. It is identical to the one obtained in the previous section—equations (3.9) and (3.14). Controller canonical form plays an important role in control theory since it represents the so-called controllable system. System controllability is one of the main concepts of modern control theory. It will be studied in detail in Chapter 5.

It is important to point out that there are many state space forms for a given dynamical system, and that all of them are related by linear transformations. More about this fact, together with the development of other important state space canonical forms, can be found in Kailath (1980; see also similarity transformation in Section 3.4).

Note that the MATLAB function `tf2ss` produces the state space form for a given transfer function, in fact, it produces the controller canonical form.

Example 3.2: The transfer function of the flexible beam from Section 2.6 is given by

$$G(s) = \frac{1.65s^4 - 0.331s^3 - 576s^2 + 90.6s + 19080}{s^6 + 0.996s^5 + 463s^4 + 97.8s^3 + 12131s^2 + 8.11s}$$

Using the direct programming technique with formulas (3.19) and (3.20), the state space controller canonical form is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -8.11 & -12131 & -97.8 & -463 & -0.996 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

and

$$y = [19080 \quad 90.6 \quad -576 \quad -0.331 \quad 1.65 \quad 0\,]\mathbf{x}$$

$\diamond$

Direct Programming Technique and Observer Canonical Form

In addition to controller canonical form, *observer canonical form* is related to another important concept of modern control theory: system observability. Observer canonical form has a very simple structure and represents an observable system. The concept of linear system observability will be considered thoroughly in Chapter 5.

Observer canonical form can be derived as follows. Equation (3.16) is written in the form

$$\begin{aligned} &Y(s)\big(s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0\big) \\ &\quad = U(s)\big(b_n s^n + b_{n-1}s^{n-1} + \cdots + b_1 s + b_0\big) \end{aligned} \tag{3.21}$$

and expressed as

$$\begin{aligned} Y(s) = &-\frac{1}{s^n}\big(a_{n-1}s^{n-1} + \cdots + a_1 s + a_0\big)Y(s) \\ &+\frac{1}{s^n}U(s)\big(b_n s^n + b_{n-1}s^{n-1} + \cdots + b_1 s + b_0\big) \end{aligned} \tag{3.22}$$

leading to

$$\begin{aligned} Y(s) = &-\frac{1}{s}a_{n-1}Y(s) - \frac{1}{s^2}a_{n-2}Y(s) - \cdots - \frac{1}{s^{n-1}}a_1 Y(s) - \frac{1}{s^n}a_0 Y(s) \\ &+ b_n U(s) + \frac{1}{s}b_{n-1}U(s) + \frac{1}{s^2}b_{n-2}U(s) + \cdots + \frac{1}{s^{n-1}}b_1 U(s) + \frac{1}{s^n}b_0 U(s) \end{aligned} \tag{3.23}$$

This relationship can be implemented by using a simulation diagram composed of n integrators in a cascade, and letting the corresponding signals to pass through the specified number of integrators. For example, terms containing $1/s$ should pass through only one integrator, signals $a_{n-2}y(t)$ and $b_{n-2}u(t)$ should pass through two integrators, and so on. Finally, signals $a_0 y(t)$ and $b_0 u(t)$ should be integrated n-times, i.e. they must pass through all n integrators. The corresponding simulation diagram is given in Figure 3.3.

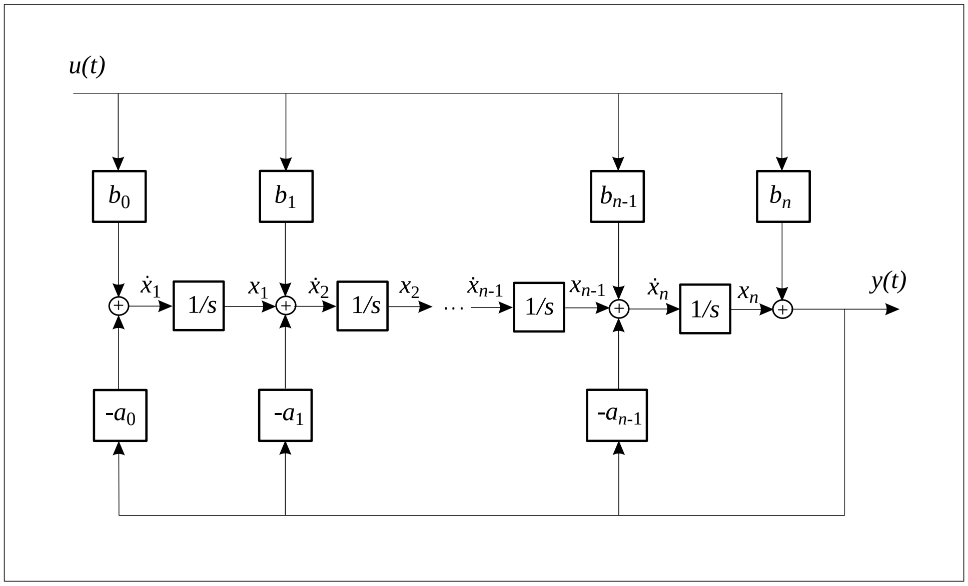

Figure 3.3: Simulation diagram for observer canonical form

Defining the state variables as the outputs of integrators, and recording relationships among state variables and the system output, we get from the above figure

$$y(t) = x_n(t) + b_n u(t) \tag{3.24}$$

$$\begin{aligned}
\dot{x}_1(t) &= -a_0 y(t) + b_0 u(t) = -a_0 x_n(t) + (b_0 - a_0 b_n)u(t) \\
\dot{x}_2(t) &= -a_1 y(t) + b_1 u(t) + x_1(t) = x_1(t) - a_1 x_n(t) + (b_1 - a_1 b_n)u(t) \\
\dot{x}_3(t) &= -a_2 y(t) + b_2 u(t) + x_2(t) = x_2(t) - a_2 x_n(t) + (b_2 - a_2 b_n)u(t) \\
&\cdots \\
\dot{x}_n(t) &= -a_{n-1} y(t) + b_{n-1} u(t) + x_{n-1}(t) \\
&= x_{n-1}(t) - a_{n-1} x_n(t) + (b_{n-1} - a_{n-1} b_n)u(t)
\end{aligned} \tag{3.25}$$

The matrix form of observer canonical form is easily obtained from (3.24) and (3.25) as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 & -a_0 \\ 1 & 0 & \dots & \dots & 0 & -a_1 \\ 0 & 1 & \ddots & \dots & \vdots & -a_2 \\ \vdots & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & -a_{n-2} \\ 0 & 0 & \dots & 0 & 1 & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} b_0 - a_0 b_n \\ b_1 - a_1 b_n \\ b_2 - a_2 b_n \\ \vdots \\ \vdots \\ b_{n-1} - a_{n-1} b_n \end{bmatrix} u(t) \quad (3.26)$$

and

$$y(t) = [0 \quad \cdots \quad 0 \quad 1\,]\mathbf{x}(t) + b_n u(t) \quad (3.27)$$

Example 3.3: The observer canonical form for the flexible beam from Example 3.2 is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -8.11 \\ 0 & 1 & 0 & 0 & 0 & -12131 \\ 0 & 0 & 1 & 0 & 0 & -97.8 \\ 0 & 0 & 0 & 1 & 0 & -463 \\ 0 & 0 & 0 & 0 & 1 & -0.996 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 19080 \\ 90.6 \\ -576 \\ -0.331 \\ 1.65 \\ 0 \end{bmatrix} u$$

and

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

◇

Observer canonical form is very useful for computer simulation of linear dynamical systems since it allows the effect of the system initial conditions to be taken into account. Thus, this form represents an observable system, in the sense to be defined in Chapter 5, which means that all state variables have an impact on the system output, and vice versa, that from the system output and state equations one is able to reconstruct the state variables at any time instant, and of course at zero, and thus, determine $x_1(0), x_2(0), ..., x_n(0)$ in terms of the original initial conditions $y(0^-), dy(0^-)/dt, ..., d^{n-1}y(0^-)/dt^{n-1}$. For more details see Section 5.5 for a subtopic on the observability role in analog computer simulation.

Parallel Programming Technique

For this technique we distinguish two cases: distinct real roots and multiple real roots of the system transfer function denominator.

Distinct Real Roots

This state space form is very convenient for applications. Derivation of this type of the model starts with the transfer function in the partial fraction expansion form. Let us assume, without loss of generality, that the polynomial in the numerator has degree of $m < n$, then

$$\begin{aligned} \frac{Y(s)}{U(s)} &= \frac{P_m(s)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \\ &= \frac{k_1}{s+p_1} + \frac{k_2}{s+p_2} + \cdots + \frac{k_n}{s+p_n} \end{aligned} \tag{3.28}$$

Here $p_1, p_2, ..., p_n$ are *distinct real* roots (poles) of the transfer function denominator.

The simulation diagram of such a form is shown in Figure 3.4.

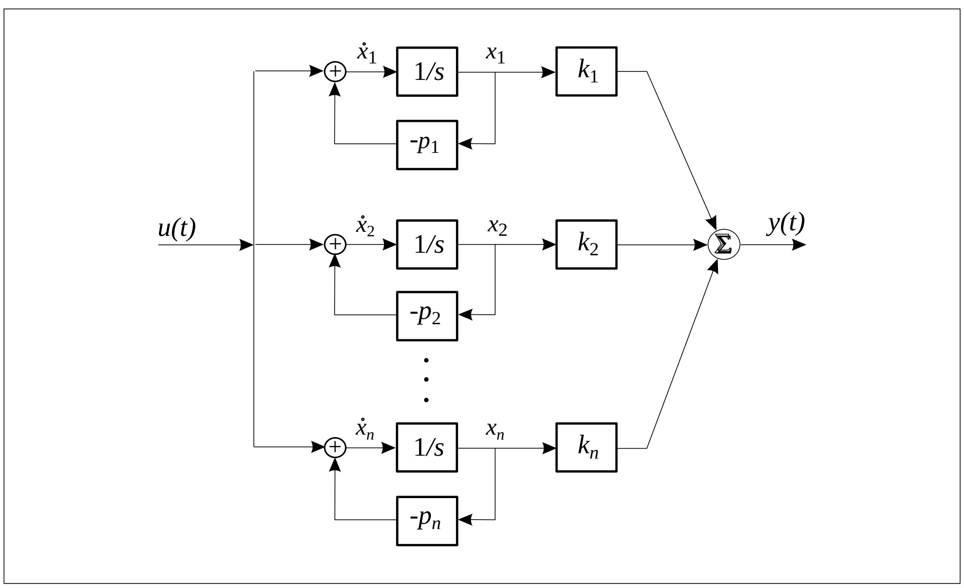

Figure 3.4: The simulation diagram for the parallel programming technique (modal canonical form)

The state space model derived from this simulation diagram is given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -p_1 & 0 & \cdots & \cdots & 0 \\ 0 & -p_2 & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & -p_n \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} u(t) \tag{3.29}$$

$$y(t) = [k_1 \quad k_2 \quad \cdots \quad \cdots \quad k_n]\mathbf{x}(t)$$

This form is known in the literature as the *modal canonical form* (also known as uncoupled form).

Example 3.4: Find the state space model of a system described by the transfer function

$$\frac{Y(s)}{U(s)} = \frac{(s+5)(s+4)}{(s+1)(s+2)(s+3)}$$

using both direct and parallel programming techniques.

The nonfactorized transfer function is

$$\frac{Y(s)}{U(s)} = \frac{s^2 + 9s + 20}{s^3 + 6s^2 + 11s + 6}$$

and the state space form obtained by using (3.19) and (3.20) of the direct programming technique is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [20 \quad 9 \quad 1]\mathbf{x}$$

Note that the MATLAB function `tf2ss` produces

$$\underline{\dot{\mathbf{x}}} = \begin{bmatrix} -6 & -11 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \underline{\mathbf{x}} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [1 \quad 9 \quad 20]\underline{\mathbf{x}}$$

which only indicates a permutation in the state space variables, that is

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \underline{\mathbf{x}}$$

Employing the partial fraction expansion (which can be obtained by the MATLAB function `residue`), the transfer function is written as

$$\frac{Y(s)}{U(s)} = \frac{(s+5)(s+4)}{(s+1)(s+2)(s+3)} = \frac{6}{s+1} - \frac{6}{s+2} + \frac{1}{s+3}$$

The state space model, directly written using (3.29), is

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 6 & -6 & 1 \end{bmatrix} \mathbf{x}$$

◇

Note that the parallel programming technique presented is valid only for the case of real distinct roots. If complex conjugate roots appear they should be combined in pairs corresponding to the second-order transfer functions, which can be independently implemented as demonstrated in the next example.

Example 3.5: Let a transfer function containing a pair of complex conjugate roots be given by

$$G(s) = \frac{4}{s+1-j} + \frac{4}{s+1+j} + \frac{2}{s+5} + \frac{3}{s+10}$$

We first group the complex conjugate poles in a second-order transfer function, that is

$$G(s) = \frac{8s+8}{s^2+2s+2} + \frac{2}{s+5} + \frac{3}{s+10}$$

Then, distinct real poles are implemented like in the case of parallel programming. A second-order transfer function, corresponding to the pair of complex conjugate poles, is implemented using direct programming, and added in parallel to the first-order transfer functions corresponding to the real poles. The simulation diagram is given in Figure 3.5, where the controller canonical form is used to represent a second-order transfer function corresponding to the complex conjugate poles. From this simulation diagram we have

$$\begin{aligned}
\dot{x}_1 &= -5x_1 + u \\
\dot{x}_2 &= -10x_2 + u \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -2x_3 - 2x_4 + u \\
y &= 2x_1 + 3x_2 + 8x_3 + 8x_4
\end{aligned}$$

so that the required state space form is

$$\dot{\mathbf{x}} = \begin{bmatrix} -5 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [2 \quad 3 \quad 8 \quad 8]\mathbf{x}$$

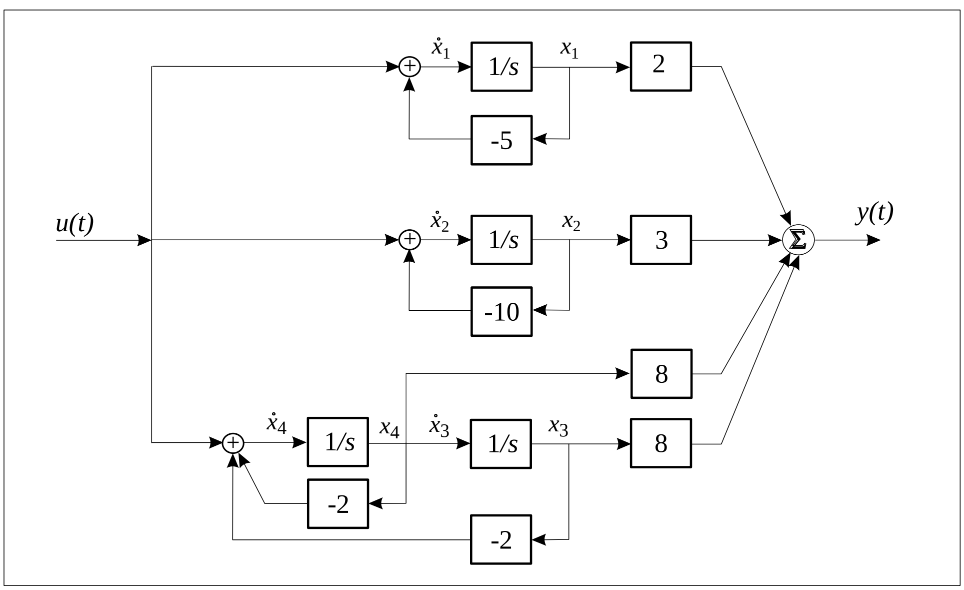

Figure 3.5: Simulation diagram for a system with complex conjugate poles

◇

Multiple Real Roots

When the transfer function has multiple real poles, it is not possible to represent the system in uncoupled form. Assume that a real pole p_1 of the transfer function has multiplicity r and that the other poles are real and distinct, that is

$$\frac{Y(s)}{U(s)} = \frac{N(s)}{(s+p_1)^r(s+p_{r+1})\cdots(s+p_n)}$$

The partial fraction form of the above expression is

$$\frac{Y(s)}{U(s)} = \frac{k_{11}}{s+p_1} + \frac{k_{12}}{(s+p_1)^2} + \cdots + \frac{k_{1r}}{(s+p_1)^r} + \frac{k_{r+1}}{s+p_{r+1}} + \cdots + \frac{k_n}{s+p_n}$$

The simulation diagram for such a system is shown in Figure 3.6.

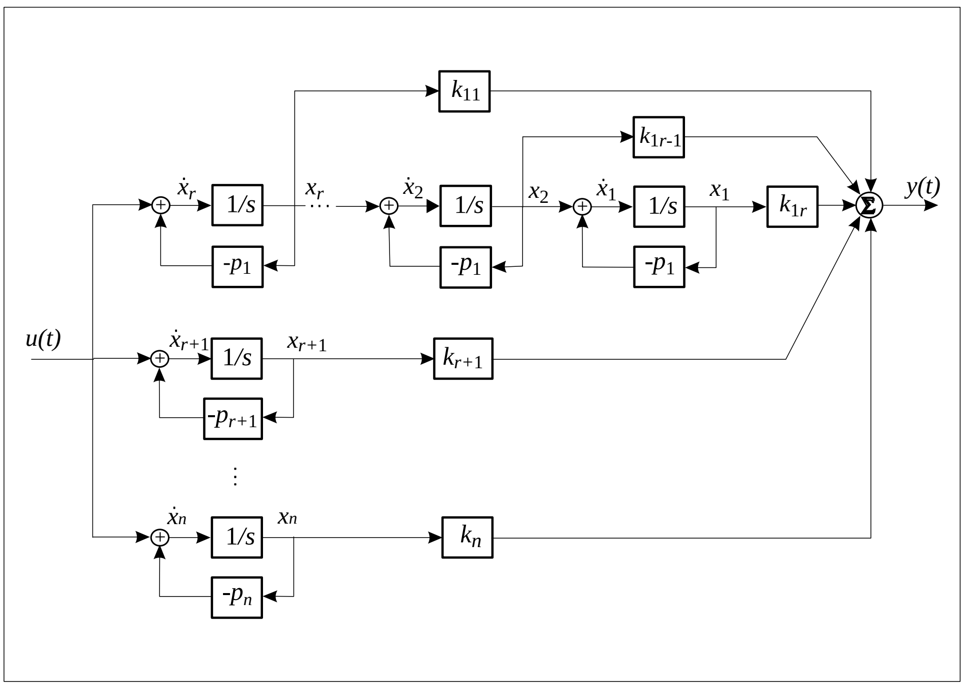

Figure 3.6: The simulation diagram for the Jordan canonical form

Taking for the state variables the outputs of integrators, the state space model is obtained as follows

$$
\mathbf{A} = \begin{bmatrix}
-p_1 & 1 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 \\
0 & -p_1 & 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & -p_1 & 1 & 0 & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & \cdots & 0 & -p_1 & 1 & 0 & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & 0 & -p_1 & 0 & \cdots & \cdots & 0 \\
0 & 0 & \cdots & \cdots & \cdots & 0 & -p_{r+1} & 0 & \cdots & 0 \\
0 & 0 & \cdots & \cdots & \cdots & 0 & \ddots & -p_{r+2} & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & \cdots & 0 & -p_n
\end{bmatrix} \tag{3.30}
$$

$$
\mathbf{B} = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & 1 & 1 & \cdots & \cdots & 1 \end{bmatrix}^T
$$

$$
\mathbf{C} = \begin{bmatrix} k_{1r} & k_{1r-1} & \cdots & \cdots & k_{12} & k_{11} & k_{r+1} & k_{r+2} & \cdots & k_n \end{bmatrix}, \quad \mathbf{D} = 0
$$

This form of the system model is known as the *Jordan canonical form*. The complete analysis of the Jordan canonical form requires a lot of space and time. However, understanding the Jordan form is crucial for correct interpretation of system stability, hence in the following chapter, the Jordan form will be completely explained.

Example 3.6: Find the state space model from the transfer function using the Jordan canonical form

$$
G(s) = \frac{s^2 + 6s + 8}{(s+1)^2(s+3)}
$$

This transfer function can be expanded as

$$
G(s) = \frac{1.25}{s+1} + \frac{1.5}{(s+1)^2} - \frac{0.25}{s+3}
$$

so that the required state space model is

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

$$
y = \begin{bmatrix} 1.5 & 1.25 & -0.25 \end{bmatrix} \mathbf{x}
$$

◇

3.2 Time Response from the State Equation

The solution of the state space equations (3.1) and (3.2) can be obtained either in the time domain by solving the corresponding matrix differential equation directly or in the frequency domain by exploiting the power of the Laplace transform. Both methods will be presented in this section.

3.2.1 Time Domain Solution

For the purpose of solving the state equation (3.1), let us first suppose that the system is in the scalar form

$$\dot{x} = ax + bu \tag{3.31}$$

with a known initial condition $x(0) = x_0$. It is very well known from the elementary theory of differential equations that the solution of (3.31) is

$$x(t) = e^{at}x_0 + \int_0^t e^{a(t-\tau)}bu(\tau)d\tau \tag{3.32}$$

The exponential term e^{at} can be expressed using the Taylor series expansion about $t_0 = 0$ as

$$e^{at} = 1 + at + \frac{1}{2!}a^2t^2 + \frac{1}{3!}a^3t^3 + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!}(at)^i \tag{3.33}$$

Analogously, in the following we prove that the solution of a general nth-order matrix state space differential equation (3.1) is given by

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.34}$$

For simplicity, we first consider the homogeneous system without control input, that is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{x}_o \tag{3.35}$$

By analogy with the scalar case, we expect the solution of this differential equation to be

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) \tag{3.36}$$

We shall prove that this is indeed a solution if (3.36) satisfies differential equation (3.35), where *the matrix exponential is defined by using the Taylor series expansion as*

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \frac{1}{3!}\mathbf{A}^3t^3 + \cdots = \sum_{i=0}^{\infty}\frac{1}{i!}\mathbf{A}^it^i \tag{3.37}$$

The proof is simple and is obtained by taking the derivative of the right-hand side of (3.37), that is

$$\frac{de^{\mathbf{A}t}}{dt} = \frac{d}{dt}\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \cdots\right)$$

$$= \mathbf{A} + \frac{2}{2!}\mathbf{A}^2t + \frac{3}{3!}\mathbf{A}^3t^2 + \cdots = \mathbf{A}\left(\mathbf{I} + \frac{1}{1!}\mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \cdots\right)$$

$$= \mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$$

Now, substitution of (3.36) into differential equation (3.35) yields

$$\dot{\mathbf{x}} = \frac{d}{dt}\mathbf{x} = \frac{d}{dt}e^{\mathbf{A}t}\mathbf{x}(0) = \mathbf{A}e^{\mathbf{A}t}\mathbf{x}(0) = \mathbf{A}\mathbf{x}(t)$$

so that matrix differential equation (3.35) is satisfied, and hence $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0)$ is its solution.

The matrix $e^{\mathbf{A}t}$ is known as the *state transition matrix* because it relates the system state at time t to that at time zero, and is denoted by

$$\Phi(t) = e^{\mathbf{A}t} \tag{3.38}$$

The state transition matrix as a time function depends only on the matrix $\mathbf{A}$. Therefore $\Phi(t)$ completely describes the internal behavior of the system, when the external influence (control input $\mathbf{u}(t)$) is absent. The system transition matrix plays a fundamental role in the theory of linear dynamical systems. In the following, we state and verify the main properties of this matrix, which is represented in the symbolic form by $e^{\mathbf{A}t}$ and so far defined only by (3.37).

Properties of the State Transition Matrix

It can be easily verified, by taking the derivative of

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}(0)$$

that the state transition matrix satisfies the linear homogeneous state equation (3.1) with the initial condition equal to an identity matrix, that is

$$\frac{d\Phi(t)}{dt} = \mathbf{A}\Phi(t), \qquad \Phi(0) = \mathbf{I} \tag{3.39}$$

The main properties of the matrix $\Phi(t)$, which follow from (3.37) and (3.38), are as follows:

(a) $\Phi(0) = \mathbf{I}$

(b) $\Phi^{-1}(t) = \Phi(-t) \Rightarrow \Phi(t)$ is nonsingular for every t

(c) $\Phi(t_2 - t_0) = \Phi(t_2 - t_1)\Phi(t_1 - t_0)$

(d) $\Phi(t)^i = \Phi(it)$, for $i \in N$

The proofs are straightforward. Property (a) is obtained when $t = 0$ is substituted into the series expansion of $e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \cdots$.

Property (b) holds, since

$$\left(e^{\mathbf{A}t}\right)^{-1} e^{\mathbf{A}t} = \mathbf{I}$$

which after multiplication from the right by $e^{-\mathbf{A}t}$ implies

$$\left(e^{\mathbf{A}t}\right)^{-1} e^{\mathbf{0}} = e^{-\mathbf{A}t} \;\Rightarrow\; \Phi^{-1}(t) = \Phi(-t)$$

and (c) follows from

$$\begin{aligned} \Phi(t_2 - t_0) &= e^{\mathbf{A}(t_2 - t_0)} = e^{\mathbf{A}(t_2 - t_1 + t_1 - t_0)} \\ &= e^{\mathbf{A}(t_2 - t_0)} e^{\mathbf{A}(t_1 - t_0)} = \Phi(t_2 - t_1)\Phi(t_1 - t_0) \end{aligned}$$

Property (d) is proved by using the fact that

$$\Phi(t)^i = \left(e^{\mathbf{A}t}\right)^i = e^{\mathbf{A}(it)} = \Phi(it)$$

In addition to properties (a), (b), (c), and (d), we have already established one additional property, namely the derivative property, as

(e)
$$\frac{d}{dt}\Phi(t) = \mathbf{A}\Phi(t) \;\Leftrightarrow\; \frac{d}{dt}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A} = \mathbf{A}e^{\mathbf{A}t}$$

The state transition matrix $\Phi(t)$ can be found by using several methods. Two of them are given in this chapter—formulas (3.37) and (3.49). The third one, very popular in linear algebra, is based on the Cayley–Hamilton theorem and is given in Appendix C.

In the case when the control vector $\mathbf{u}(t)$ is present in the system (forced response)

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{x}(0) = \mathbf{x}_{\mathrm{o}}$$

we look for the solution of the state space equation in the form

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{f}(t) \tag{3.40}$$

Then

$$\dot{\mathbf{x}}(t) = \mathbf{A}e^{\mathbf{A}t}\mathbf{f}(t) + e^{\mathbf{A}t}\dot{\mathbf{f}}(t) = \mathbf{A}\mathbf{x} + e^{\mathbf{A}t}\dot{\mathbf{f}} \tag{3.41}$$

It follows from (3.1) and (3.41) that

$$e^{\mathbf{A}t}\dot{\mathbf{f}}(t) = \mathbf{B}\mathbf{u} \tag{3.42}$$

From (3.42) we have

$$\dot{\mathbf{f}}(t) = \left(e^{\mathbf{A}t}\right)^{-1}\mathbf{B}\mathbf{u} = e^{-\mathbf{A}t}\mathbf{B}\mathbf{u} \tag{3.43}$$

Integrating this equation, bearing in mind that $\mathbf{x}(0) = e^{\mathbf{A}\cdot 0}\mathbf{f}(0) = \mathbf{f}(0)$, we get

$$\mathbf{f}(t) - \mathbf{f}(0) = \int_0^t e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.44}$$

Substitution of the last expression in (3.40) gives the required solution

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.45}$$

or

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}(0) + \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.46}$$

When the initial state of the system is known at time t_0, rather than at time $t = 0$, the solution of the state equation is similarly obtained as

$$\begin{aligned}\mathbf{x}(t) &= \Phi(t-t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \\ &= e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau\end{aligned} \tag{3.47}$$

This can be easily verified by repeating steps (3.40)–(3.45) with $\mathbf{x}(t_0) = \mathbf{x}_o$ and $\mathbf{x}(t_0) = e^{\mathbf{A}t_0}\mathbf{f}(t_0)$.

Example 3.7: For the system given in Example 3.4 find the state transition matrix $\Phi(t)$. Evaluate $\Phi(1)$ using the MATLAB function `expm`. Assuming that the initial state of the system is zero, find the state response to a unit step. Check the solution obtained by using the MATLAB function `step`.

At the present time we are able to find the state transition matrix (matrix exponential) by using formula (3.37), which deals with an infinite series, and hence is not very convenient for calculations. Better ways to find $\Phi(t)$ are the method based on the Cayley–Hamilton theorem (see Appendix C) and the formula based on the Laplace transform, see formula (3.49). However, in this problem, if we start with the uncoupled (modal) state space form of the system considered in Example 3.4, we can avoid using any of the above methods in order to find the state transition matrix. Namely, *for diagonal matrices only*, it is easy to show that

$$\Phi(t) = e^{\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & -3\end{bmatrix}t} = \begin{bmatrix}e^{-t} & 0 & 0\\ 0 & e^{-2t} & 0\\ 0 & 0 & e^{-3t}\end{bmatrix}$$

Using the MATLAB function for evaluating the matrix exponential as `expm(A*1)`, we get

$$\Phi(1) = \begin{bmatrix}e^{-1} & 0 & 0\\ 0 & e^{-2} & 0\\ 0 & 0 & e^{-3}\end{bmatrix} = \begin{bmatrix}0.3679 & 0 & 0\\ 0 & 0.1353 & 0\\ 0 & 0 & 0.0498\end{bmatrix}$$

The state response to a unit step is computed from (3.46) as

$$\mathbf{x}(t) = \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau = \int_0^t \begin{bmatrix} e^{-(t-\tau)} & 0 & 0 \\ 0 & e^{-2(t-\tau)} & 0 \\ 0 & 0 & e^{-3(t-\tau)} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot 1 d\tau$$

$$= \int_0^t \begin{bmatrix} e^{-(t-\tau)} \\ e^{-2(t-\tau)} \\ e^{-3(t-\tau)} \end{bmatrix} d\tau = \begin{bmatrix} 1 - e^{-t} \\ 0.5\left(1 - e^{-2t}\right) \\ 0.333\left(1 - e^{-3t}\right) \end{bmatrix}$$

The step responses of system states, obtained by using MATLAB statements `[y,x]=step(A,B,C,D)` and `plot(x)`, with

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 6 & -6 & 1 \end{bmatrix}, \ \mathbf{D} = 0$$

are shown in Figure 3.7.

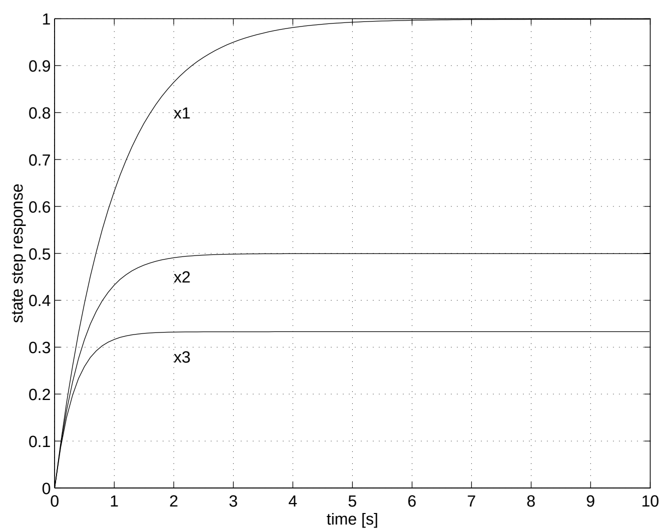

Figure 3.7: The state responses for Example 3.7

◇

3.2.2 Solution Using the Laplace Transform

The time trajectory of the state vector $\mathbf{x}(t)$ can be also found using the Laplace transform method. The main properties of the Laplace transform and common transform pairs are given in Appendix A.

The Laplace transform applied to the state equation (3.1) gives

$$s\mathbf{X}(s) - \mathbf{x}(0^-) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s)$$

or

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0^-) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \tag{3.48}$$

Let us assume that $\mathbf{x}(0) = \mathbf{x}(0^-)$. Comparing equations (3.46) and (3.48), it is easy to see that the term $(s\mathbf{I} - \mathbf{A})^{-1}$ is the Laplace transform of the state transition matrix, that is

$$\Phi(s) = (s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{\det(s\mathbf{I} - \mathbf{A})}\mathrm{adj}(s\mathbf{I} - \mathbf{A}) = \mathcal{L}\{\Phi(t)\} \tag{3.49}$$

The time form of the state vector $\mathbf{x}(t)$ is obtained by applying the inverse Laplace transform to

$$\mathbf{X}(s) = \Phi(s)\mathbf{x}(0) + \Phi(s)\mathbf{B}\mathbf{U}(s) \tag{3.50}$$

Note that the second term on the right-hand side corresponds in the time domain to the convolution integral, so that we have

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.51}$$

Once the state vector $\mathbf{x}(t)$ is determined, the output vector $\mathbf{y}(t)$ of the system is simply obtained by substitution of $\mathbf{x}(t)$ into equation (3.2), that is

$$\mathbf{y}(t) = \mathbf{C}\Phi(t)\mathbf{x}(0) + \mathbf{C}\int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau + \mathbf{D}\mathbf{u}(t) \tag{3.52}$$

or, in the complex domain

$$\mathbf{Y}(s) = \mathbf{C}\Phi(s)\mathbf{x}(0) + [\mathbf{C}\Phi(s)\mathbf{B} + \mathbf{D}]\mathbf{U}(s) \tag{3.53}$$

3.2.3 State Space Model and Transfer Function

The matrix that establishes a relationship between the output vector $\mathbf{Y}(s)$ and the input vector $\mathbf{U}(s)$, for the zero initial conditions, $\mathbf{x}(0) = 0$, is called the *system matrix transfer function*. From (3.53) it is given by

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{3.54}$$

Note that (3.54) represents the open-loop system matrix transfer function.

Example 3.8: Find the transfer function for the system given in Example 3.4.

It is the easiest to use modal (uncoupled) canonical form, which leads to

$$\begin{aligned} G(s) &= [6 \quad -6 \quad 1] \begin{bmatrix} s+1 & 0 & 0 \\ 0 & s+2 & 0 \\ 0 & 0 & s+3 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= [6 \quad -6 \quad 1] \begin{bmatrix} \frac{1}{s+1} & 0 & 0 \\ 0 & \frac{1}{s+2} & 0 \\ 0 & 0 & \frac{1}{s+3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \frac{6}{s+1} - \frac{6}{s+2} + \frac{1}{s+3} = \frac{(s+5)(s+4)}{(s+1)(s+2)(s+3)} \end{aligned}$$

If we start with controller canonical form we will get, after some algebra,

$$G(s) = [20 \quad 9 \quad 1] \begin{bmatrix} s+6 & -1 & 0 \\ 11 & s & -1 \\ 6 & 0 & s \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= [20 \quad 9 \quad 1] \frac{1}{s^3 + 6s^2 + 11s + 6} \begin{bmatrix} s^2 + 6s + 11 & s+6 & 1 \\ -6 & s(s+6) & s \\ -6s & -11s - 6 & s^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \frac{s^2 + 9s + 20}{s^3 + 6s^2 + 11s + 6}$$

Note that the MATLAB function `ss2tf` can be used to solve the above problem.

◇

3.3 Discrete-Time Models

Discrete-time systems are either inherently discrete (e.g. models of bank accounts, national economy growth models, population growth models, digital words) or they are obtained as a result of sampling (discretization) of continuous-time systems. In such kinds of systems, inputs, state space variables, and outputs have the discrete form and the system models can be represented in the form of transition tables.

The mathematical model of a discrete-time system can be written in terms of a recursive formula by using linear matrix difference equations as

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \mathbf{A}_d\mathbf{x}(kT) + \mathbf{B}_d\mathbf{u}(kT) \\ \mathbf{y}(kT) &= \mathbf{C}_d\mathbf{x}(kT) + \mathbf{D}_d\mathbf{u}(kT) \end{aligned} \tag{3.55}$$

Here T represents the sampling interval, which may be omitted for brevity. Even more, in the case of inherent discrete systems, there is no need to introduce the notion of the sampling interval T so that these systems are described by (3.55) with $T = 1$.

Similarly to continuous-time systems, discrete state space equations can be derived either from difference equations (Subsection 3.3.1) or from discrete transfer functions using simulation diagrams (Subsection 3.3.2). In Subsection 3.3.3 we show how to discretize continuous-time linear systems and obtain discrete-time ones. In the rest of the section we parallel most of the results obtained in previous sections for continuous-time systems.

3.3.1 Difference Equations and State Space Form

An nth-order difference equation is given by

$$\begin{aligned} & y(k+n) + a_{n-1}y(k+n-1) + \cdots + a_1y(k+1) + a_0y(k) \\ & \quad = b_nu(k+n) + b_{n-1}u(k+n-1) + \cdots + b_1u(k+1) + b_0u(k) \end{aligned} \tag{3.56}$$

This equation expresses all values in terms of discrete-time k.

The corresponding state space equation can be derived by using the same techniques as in the continuous-time case. For example, for phase variable canonical form (controller canonical form) in discrete-time, we have

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} (b_0 - a_0 b_n) & (b_1 - a_1 b_n) & \cdots & (b_{n-1} - a_{n-1} b_n) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_n u(k) \tag{3.57}$$

Note that the transformation equations, dual to the continuous-time ones (3.11)–(3.13), are given in the discrete-time domain by

$$\xi(k+n) + a_{n-1}\xi(k+n-1) + \cdots + a_1\xi(k+1) + a_0\xi(k) = u(k) \tag{3.58}$$

$$\begin{aligned} x_1(k) &= \xi(k) \\ x_2(k) &= \xi(k+1) \\ x_3(k) &= \xi(k+2) \\ &\vdots \\ x_n(k) &= \xi(k+n-1) \end{aligned} \tag{3.59}$$

$$y(k) = b_0\xi(k) + b_1\xi(k+1) + b_2\xi(k+2) + \cdots + b_n\xi(k+n) \tag{3.60}$$

Eliminating $\xi(k+n)$ from (3.60) by using (3.58) and (3.59), the output equation given in (3.57) is obtained.

3.3.2 Discrete Transfer Function and State Space Model

The derivation of state space equations from discrete transfer functions, based on simulation diagrams, is very similar to the continuous-time case. The only difference is that in simulation diagrams the integration block $1/s$ is replaced by the unit delay element z^{-1}. *The state variables are selected as outputs of these delay elements*. We shall illustrate this method by an example.

Example 3.9: Find two state space forms for the transfer function

$$\frac{Y(z)}{U(z)} = \frac{z + 1.1}{(z - 0.9)(z + 0.7)(z - 0.7)}$$

We solve this problem by using both direct (a) and parallel (b) programming techniques.

(a) The transfer function can be rewritten as

$$\frac{Y(z)}{U(z)} = \frac{z + 1.1}{z^3 - 0.9z^2 - 0.49z + 0.441}$$

The simulation diagram for this transfer function is shown in Figure 3.8.

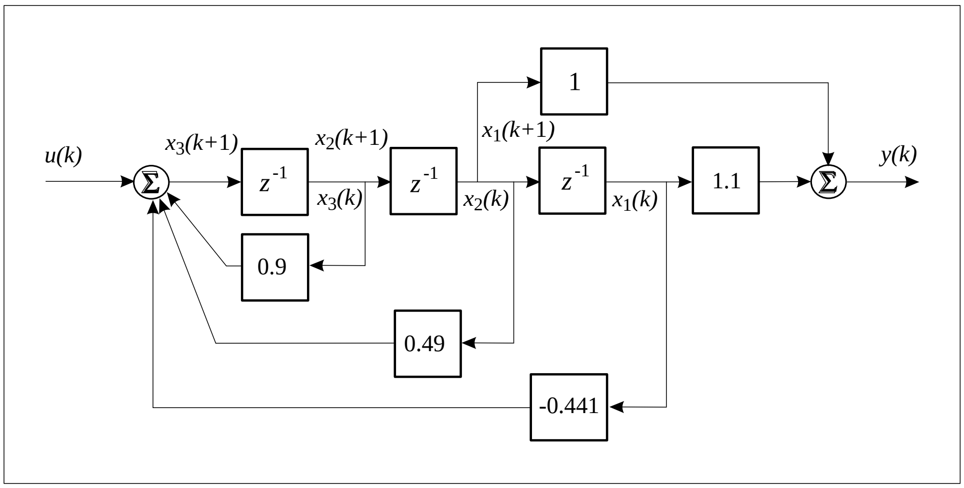

Figure 3.8: Simulation diagram for direct programming in discrete-time domain (controller canonical form)

The state space model is obtained from this simulation diagram by using the outputs of delay elements as state variables. It is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.441 & 0.49 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1.1 \quad 1 \quad 0]\mathbf{x}(k)$$

Note that controller canonical form could have been obtained without drawing the simulation diagram. We know that this form is identical to phase variable canonical form, which is represented by (3.57). Identifying the corresponding coefficients in the original transfer function, the desired state space form is obtained directly from (3.57). We have used the above method in order to demonstrate at the same time the procedure of drawing simulation diagrams in the discrete-time domain.

(b) Employing the partial fraction expansion (with help of the MATLAB function `residue`), we get

$$\frac{Y(z)}{U(z)} = \frac{6.25}{z-0.9} + \frac{0.1786}{z+0.7} + \frac{-6.4286}{z-0.7}$$

Since the poles of the transfer function are real and distinct we get the modal canonical form as

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & -0.7 & 0 \\ 0 & 0 & 0.7 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [6.25 \quad 0.1786 \quad -6.4286]\mathbf{x}(k)$$

◇

3.3.3 Discretization of Continuous-Time Systems

Real physical dynamic systems are continuous in nature. In this section, we show how to obtain discrete-time state space models from continuous-time system models. Assume that the plant is linear, continuous, and time invariant with r-inputs and p-outputs (see Figure 3.9). Inputs are sampled by using the zero-order hold (ZOH) device. This device samples inputs at discrete-time instants kT (see Figure 3.10b) and the values obtained for vector $\mathbf{u}(kT)$ are held until $(k+1)T$. The corresponding signal is given in Figure 3.10c.

The state space model of such a plant is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{m}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{m}(t) \end{aligned} \tag{3.61}$$

These equations define states and outputs during the sampling interval $kT \leq t < (k+1)T$. Input signals $m_i(t), i = 1, ..., r$, are defined by

$$m_i(t) = m_i(kT) = u_i(kT), \quad kT \leq t < (k+1)T, \quad k = 0, 1, 2, \ldots \tag{3.62}$$

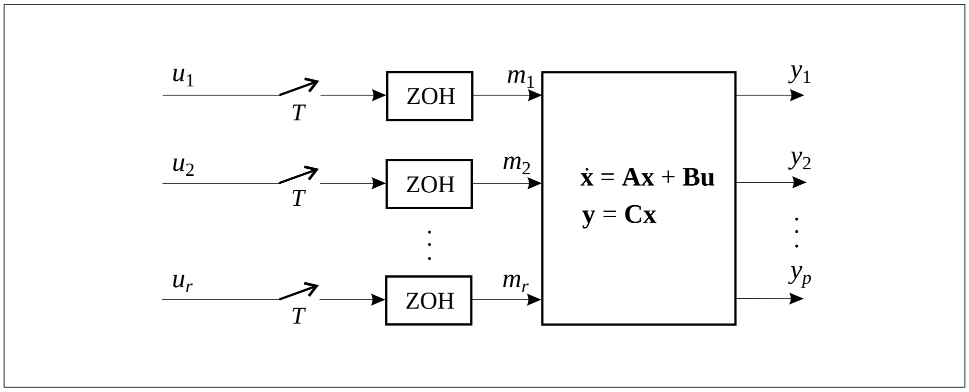

Figure 3.9: Sampling in a multivariable controlled plant

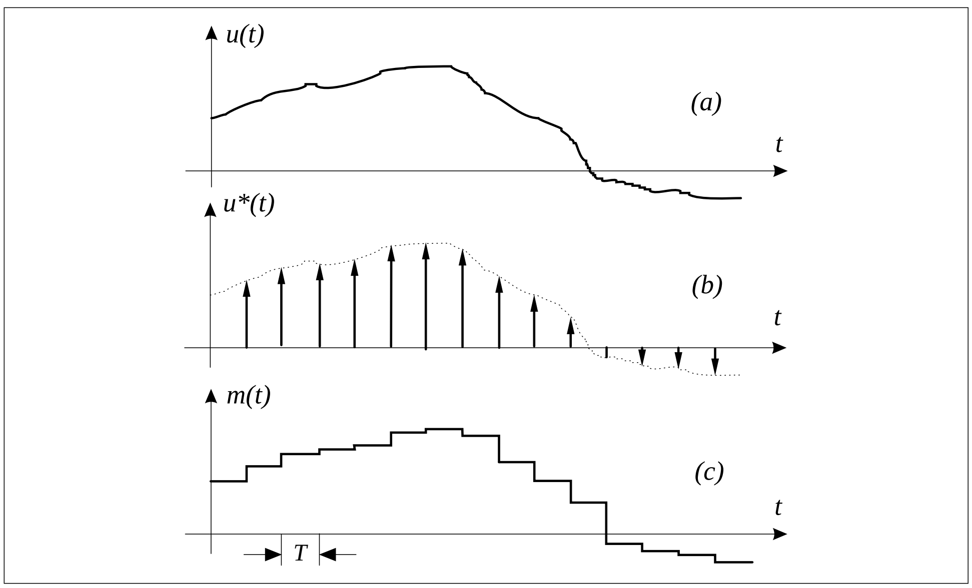

Figure 3.10: Transformation of a continuous-time input signal by the zero-order hold element

In the following, we show how to perform discretization of a continuous-time state space model (3.61) and obtain a discrete-time state space model having the form of (3.55) together with the corresponding expressions for matrices $\mathbf{A}_d, \mathbf{B}_d, \mathbf{C}_d,$ and $\mathbf{D}_d$.

Consider formula (3.45) with $t = T$

$$\mathbf{x}(T) = e^{\mathbf{A}T}\mathbf{x}(0) + \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{B}\mathbf{u}(0)d\tau$$

$$= e^{\mathbf{A}T}\mathbf{x}(0) + e^{\mathbf{A}T}\int_0^T e^{-\mathbf{A}\tau}d\tau\mathbf{B}\mathbf{u}(0) = \Phi(T)x(0) + \int_0^T \Phi(T-\tau)d\tau\mathbf{B}\mathbf{u}(0) \tag{3.63}$$

which can be written in the form

$$\mathbf{x}(T) = \mathbf{A}_d\mathbf{x}(0) + \mathbf{B}_d\mathbf{u}(0) \tag{3.64}$$

Comparing (3.63) and (3.64) we can find expressions for $\mathbf{A}_d$ and $\mathbf{B}_d$. They are given by

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}T} = \Phi(T) \\ \mathbf{B}_d &= e^{\mathbf{A}T}\int_0^T e^{-\mathbf{A}\tau}d\tau\mathbf{B} = \int_0^T e^{\mathbf{A}(T-\tau)}d\tau \cdot \mathbf{B} = \int_0^T e^{\mathbf{A}\sigma}d\sigma \cdot \mathbf{B} \end{aligned} \tag{3.65}$$

Note that $\mathbf{A}_d$ and $\mathbf{B}_d$ are obtained for the time interval from 0 to T. However, it can easily be shown that due to system time invariance the same expressions for $\mathbf{A}_d$ and $\mathbf{B}_d$ are obtained for any time interval. Namely, steps (3.63)–(3.65) can be repeated for succeeding time intervals $2T, 3T, ..., (k+1)T$ with initial conditions taken, respectively, as $\mathbf{x}(T), \mathbf{x}(2T), ..., \mathbf{x}(kT)$. Therefore, for the time instant $t = (k+1)T$ and for $t_0 = kT$, we have from (3.47)

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \Phi[(k+1)T - kT]\mathbf{x}(kT) + \int_{kT}^{(k+1)T} \Phi[(k+1)T - \tau]d\tau\mathbf{B}\mathbf{u}(kT) \\ &= \mathbf{A}_d\mathbf{x}(kT) + \mathbf{B}_d\mathbf{u}(kT) \end{aligned} \tag{3.66}$$

From the above equation we see that the matrices $\mathbf{A}_d$ and $\mathbf{B}_d$ are given by

$$\begin{aligned} \mathbf{A}_d &= \Phi[(k+1)T - kT] = \Phi(T) = e^{\mathbf{A}T} \\ \mathbf{B}_d &= \int_{kT}^{(k+1)T} \Phi[(k+1)T - \tau]d\tau\mathbf{B} = \int_0^T \Phi(\sigma)d\sigma\mathbf{B} = \int_0^T e^{\mathbf{A}\sigma}d\sigma\mathbf{B} \end{aligned} \tag{3.67}$$

The last equality is obtained by using change of variables as $\sigma = (k+1)T - \tau$. Since (3.65) and (3.67) are identical, we conclude that for a time invariant continuous-time linear system, the discretization procedure yields a time invariant discrete-time linear system whose matrices $\mathbf{A}_d$ and $\mathbf{B}_d$ depend only on $\mathbf{A}$, $\mathbf{B}$, and the sampling interval T.

In a similar manner the output equation (3.61) at $t = kT$ is given by

$$\mathbf{y}(kT) = \mathbf{C}\mathbf{x}(kT) + \mathbf{D}\mathbf{u}(kT) \tag{3.68}$$

Comparing this equation with the general output equation of linear discrete-time systems (3.55), we conclude that

$$\mathbf{C}_d = \mathbf{C}, \quad \mathbf{D}_d = \mathbf{D} \tag{3.69}$$

In the literature one can find several methods for discretization of continuous-time linear systems. The discretization technique presented in this section is known as the *integral approximation method*.

In the case of discrete-time linear systems obtained by sampling continuous-time linear systems, the matrix $\mathbf{A}_d$, given by (3.65), can be determined from the infinite series

$$\mathbf{A}_d = e^{\mathbf{A}T} = \mathbf{I} + \mathbf{A}T + \frac{1}{2!}\mathbf{A}^2T^2 + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!}\mathbf{A}^iT^i \tag{3.70}$$

It can be also obtained either by using formula (3.49) or the method based on the Cayley–Hamilton theorem and setting $t = T$ in $\Phi(t) = e^{\mathbf{A}t}$. Also, in order to evaluate $e^{\mathbf{A}T}$ we can use MATLAB function `expm(A*T)`.

To find $\mathbf{B}_d$, the second expression in (3.65) is integrated to give (see Appendix C—matrix integrals)

$$\mathbf{B}_d = e^{\mathbf{A}T}\left(-e^{-\mathbf{A}T}\mathbf{A}^{-1} + \mathbf{A}^{-1}\right)\mathbf{B} = (\mathbf{A}_d - \mathbf{I})\mathbf{A}^{-1}\mathbf{B} \tag{3.71}$$

which is valid under the assumption that $\mathbf{A}$ is invertible. If $\mathbf{A}$ is singular, $\mathbf{B}_d$ can be determined from

$$\mathbf{B}_d = \left(\sum_{i=1}^{\infty} \frac{T^i}{i!}\mathbf{A}^{i-1}\right)\mathbf{B} = \left(\sum_{i=0}^{\infty} \frac{T^{i+1}}{(i+1)!}\mathbf{A}^i\right)\mathbf{B} \tag{3.72}$$

which is obtained by using (3.37) in (3.67) and performing the corresponding integration. Note that the above sum converges quite quickly so that only a few terms give quite an accurate expression for $\mathbf{B}_d$.

Example 3.10: Find the discrete-time state space model of a continuous-time system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}$$

The sampling period T is equal to 0.1.

According to (3.65) and (3.69), we have from (3.49)

$$\mathbf{A}_d = \Phi(T) = \begin{bmatrix} 2e^{-T} - e^{-2T} & e^{-T} - e^{-2T} \\ 2e^{-2T} - 2e^{-T} & 2e^{-2T} - e^{-T} \end{bmatrix} = \begin{bmatrix} 0.9909 & 0.0861 \\ -0.1722 & 0.7326 \end{bmatrix}$$

$$\mathbf{B}_d = (\mathbf{A}_d - \mathbf{I})\mathbf{A}^{-1}\mathbf{B} = \begin{bmatrix} \frac{1}{2}\left(1 + e^{-2T}\right) - e^{-T} \\ e^{-T} - e^{-2T} \end{bmatrix} = \begin{bmatrix} 0.0045 \\ 0.0861 \end{bmatrix}$$

$$\mathbf{C}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D}_d = 0$$

The same result is obtained by using the MATLAB function for discretization of a continuous state space model as `[Ad,Bd]=c2d(A,B,T)`.

◇

3.3.4 Solution of the Discrete-Time State Equation

The objective of this section is to find the solution of the difference state equation (3.55) for the given initial state $\mathbf{x}(0)$ and the control signal $\mathbf{u}(k)$ at the sampling instants $T, 2T, \ldots, kT$. For simplicity we assume $T = 1$.

From the state equation $\mathbf{x}(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_d \mathbf{u}(k)$, for $k = 0, 1, \ldots,$ $N - 1$, it follows

$$\mathbf{x}(1) = \mathbf{A}_d \mathbf{x}(0) + \mathbf{B}_d \mathbf{u}(0)$$

$$\mathbf{x}(2) = \mathbf{A}_d \mathbf{x}(1) + \mathbf{B}_d \mathbf{u}(1) = \mathbf{A}_d^2 \mathbf{x}(0) + \mathbf{A}_d \mathbf{B}_d \mathbf{u}(0) + \mathbf{B}_d \mathbf{u}(1)$$

$$\mathbf{x}(3) = \mathbf{A}_d \mathbf{x}(2) + \mathbf{B}_d \mathbf{u}(2) = \mathbf{A}_d^3 \mathbf{x}(0) + \mathbf{A}_d^2 \mathbf{B}_d \mathbf{u}(0) + \mathbf{A}_d \mathbf{B}_d \mathbf{u}(1) + \mathbf{B}_d \mathbf{u}(2)$$

$$\vdots$$

$$\mathbf{x}(N) = \mathbf{A}_d\mathbf{x}(N-1) + \mathbf{B}_d\mathbf{u}(N-1) = \mathbf{A}_d^N\mathbf{x}(0) + \sum_{i=0}^{N-1}\mathbf{A}_d^{N-i-1}\mathbf{B}_d\mathbf{u}(i) \tag{3.73}$$

Using the notion of the *discrete-time state transition matrix* defined by

$$\Phi_d(k) = \mathbf{A}_d^k \tag{3.74}$$

we get

$$\mathbf{x}(N) = \Phi_d(N)\mathbf{x}(0) + \sum_{i=0}^{N-1}\Phi_d(N-i-1)\mathbf{B}_d\mathbf{u}(i) \tag{3.75}$$

Note that the discrete-time state transition matrix relates the state of an input-free system at initial time ($k = 0$) to the state of the system at any other time $k > 0$, that is

$$\mathbf{x}(k) = \Phi_d(k)\mathbf{x}(0) = \mathbf{A}_d^k\mathbf{x}(0) \tag{3.76}$$

It is easy to verify that the discrete-time state transition matrix has the following properties

(a) $\Phi_d(0) = \mathbf{A}_d^0 = \mathbf{I} \;\Leftarrow\; \mathbf{x}(0) = \Phi_d(0)\mathbf{x}(0)$

(b) $\Phi_d(k_2 - k_0) = \Phi_d(k_2 - k_1)\Phi_d(k_1 - k_0) = \mathbf{A}_d^{k_2-k_1}\mathbf{A}_d^{k_1-k_0} = \mathbf{A}_d^{k_2-k_0}$

(c) $\Phi_d^i(k) = \Phi_d(ik) \;\Leftarrow\; \left(\mathbf{A}_d^k\right)^i = \mathbf{A}_d^{ik}$

(d) $\Phi_d(k+1) = \mathbf{A}_d\Phi_d(k), \quad \Phi_d(0) = \mathbf{I}$

The last property follows from

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) \;\Rightarrow\; \Phi_d(k+1)\mathbf{x}(0) = \mathbf{A}_d\Phi_d(k)\mathbf{x}(0)$$

It is important to point out that the discrete-time state transition matrix may be singular, which follows from the fact that $\mathbf{A}_d^k$ is nonsingular if and only if the matrix $\mathbf{A}_d$ is nonsingular. In the case of inherent discrete-time systems, the matrix $\mathbf{A}_d$ may be singular in general. However, if $\mathbf{A}_d$ is obtained through the discretization procedure of a continuous-time linear system, like in (3.65), then

$$(\mathbf{A}_d)^{-1} = \left(e^{\mathbf{A}T}\right)^{-1} = e^{-\mathbf{A}T}$$

so that the discrete-time state transition matrix is nonsingular in this case.

Remark 1: If the initial value of the state vector is not $\mathbf{x}(0)$ but $\mathbf{x}(k_0)$, then the solution (3.75) is

$$\mathbf{x}(k_0+N) = \Phi_d(N)\mathbf{x}(k_0) + \sum_{i=0}^{N-1} \Phi_d(N-i-1)\mathbf{B}_d\mathbf{u}(k_0+i) \tag{3.77}$$

The output of the system at sampling instant $k = N$ is obtained by substituting $\mathbf{x}(k)$ from (3.75) into the output equation, producing

$$\mathbf{y}(N) = \mathbf{C}_d\Phi_d(N)\mathbf{x}(0) + \mathbf{C}_d \sum_{i=0}^{N-1} \Phi(N-i-1)\mathbf{B}_d\mathbf{u}(i) + \mathbf{D}_d\mathbf{u}(N) \tag{3.78}$$

Note that for $T \neq 1$, equations (3.75) and (3.78) are modified as

$$\mathbf{x}(NT) = \Phi_d(NT)\mathbf{x}(0) + \sum_{i=0}^{N-1} \Phi_d[(N-i-1)T]\mathbf{B}_d\mathbf{u}(iT) \tag{3.79}$$

$$\mathbf{y}(NT) = \mathbf{C}_d\Phi_d(NT)\mathbf{x}(0) + \mathbf{C}_d \sum_{i=0}^{N-1} \Phi[(N-i-1)T]\mathbf{B}_d\mathbf{u}(iT) + \mathbf{D}_d\mathbf{u}(NT) \tag{3.80}$$

Remark 2: The discrete-time state transition matrix defined by $\mathbf{A}_d^k$ can be evaluated efficiently for large values of k by using a method based on the Cayley–Hamilton theorem and described in Appendix C. It can be also evaluated by using the $\mathcal{Z}$-transform method as given in formula (3.85), see Subsection 3.3.5.

Example 3.11: For the system given in Example 3.10, use MATLAB to find the unit step and impulse responses assuming that the initial condition is $\mathbf{x}(0) = [0 \quad 0]^T$.

The required time responses can be obtained directly by using MATLAB statements `[y,x]=dstep(Ad,Bd,Cd,Dd)` (for step response) and `[y,x]=dimpulse(Ad,Bd,Cd,Dd)` (for impulse response) with the discrete-time model matrices obtained in the last example. The corresponding state and output responses are presented in Figure 3.11.

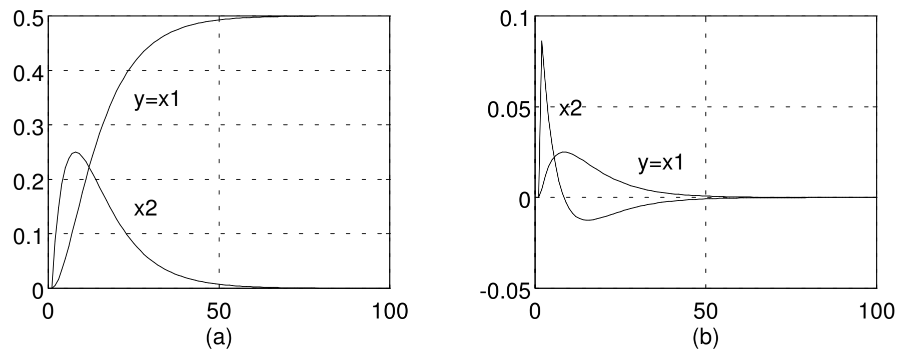

Figure 3.11: (a) Step responses; (b) impulse responses

The same problem could have been solved analytically as follows. Since the initial condition is zero and $u(k) = 1$ for $k \geq 0$, we get from (3.73) the state response as

$$\mathbf{x}(N) = \sum_{i=0}^{N-1} \mathbf{A}_d^{N-i-1}\mathbf{B}_d, \quad N = 1, 2, ...$$

The output response, obtained from (3.78), is given by

$$y(N) = \mathbf{C}\mathbf{x}(N), \quad N = 1, 2, ...$$

However, at this point, for large N one is faced with the problem of efficiently calculating the powers of matrix $\mathbf{A}_d$. This can be facilitated analytically by using either the Cayley–Hamilton theorem (see Appendix C) or the $\mathcal{Z}$-transform method to be presented in the next subsection.

By the Cayley–Hamilton method, we have for a 2×2 matrix that

$$\mathbf{A}_d^k = \alpha_0\mathbf{I} + \alpha_1\mathbf{A}_d, \quad k = 2, 3, 4, ...$$

with α_0 and α_1 satisfying

$$\begin{aligned}\lambda_1^k &= \alpha_0 + \alpha_1\lambda_1\\ \lambda_2^k &= \alpha_0 + \alpha_1\lambda_2\end{aligned}$$

where λ_1 and λ_2 are distinct eigenvalues of $\mathbf{A}_d$. System eigenvalues will be considered in Section 3.4.

◇

3.3.5 Solution of the Discrete State Equation by the $\mathcal{Z}$-transform

Applying the $\mathcal{Z}$-transform (see Appendix B) to the state space equation of a discrete-time system

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \tag{3.81}$$

we get

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}_d\mathbf{X}(z) + \mathbf{B}_d\mathbf{U}(z) \tag{3.82}$$

The complex state vector $\mathbf{X}(z)$ can be expressed as

$$\mathbf{X}(z) = (z\mathbf{I} - \mathbf{A}_d)^{-1}z\mathbf{x}(0) + (z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d\mathbf{U}(z) \tag{3.83}$$

The inverse z-transform of the last equation gives $\mathbf{x}(k)$, that is

$$\mathbf{x}(k) = \mathcal{Z}^{-1}\Big[(z\mathbf{I} - \mathbf{A}_d)^{-1}z\Big]x(0) + \mathcal{Z}^{-1}\Big[(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d\mathbf{U}(z)\Big] \tag{3.84}$$

Comparing equations (3.75) and (3.84) we conclude that

$$\Phi_d(k) = \mathcal{Z}^{-1}\Big[(z\mathbf{I} - \mathbf{A}_d)^{-1}z\Big] = \mathbf{A}_d^k, \quad k = 1, 2, 3, \ldots \tag{3.85}$$

Let us repeat and emphasize that the discrete state transition matrix $\Phi_d(k)$ of a general discrete-time invariant linear system can be obtained either by using (3.85) or the Cayley–Hamilton method given in Appendix C.

The inverse transform of the second term on the right-hand side of (3.84) is obtained directly by the application of the discrete convolution theorem (see Appendix B), leading to

$$\mathcal{Z}^{-1}\Big\{(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d\mathbf{U}(z)\Big\} = \sum_{i=0}^{k-1}\Phi_d(k-1-i)\mathbf{B}_d\mathbf{u}(i) \tag{3.86}$$

Combining (3.84) and (3.86) we get the required solution of the discrete-time state space equation as

$$\mathbf{x}(k) = \Phi_d(k)\mathbf{x}(0) + \sum_{i=0}^{k-1} \Phi_d(k-i-1)\mathbf{B}_d\mathbf{u}(i) \tag{3.87}$$

The complex form of the output vector $\mathbf{Y}(z)$ is obtained if the $\mathcal{Z}$-transform is applied to the output equation

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

and $\mathbf{X}(z)$ is substituted from (3.83), leading to

$$\mathbf{Y}(z) = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}z\mathbf{x}(0) + \left[\mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d\right]\mathbf{U}(z)$$

From the above expression, for the zero initial condition, i.e. $\mathbf{x}(0) = 0$, the *discrete matrix transfer function* is given by

$$\mathbf{G}_d(z) = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d \tag{3.88}$$

3.3.6 Response Between Sampling Instants

An important feature of the state variable method is that it can be modified to determine the output between sampling instants. Let $t_0 = kT$ and $t = (k+\Delta)T$, where $0 \leq \Delta < 1$. Equation (3.47) gives

$$\mathbf{x}[(k+\Delta)T] = e^{\mathbf{A}\Delta T}\mathbf{x}(kT) + \int_{kT}^{(k+\Delta)T} e^{\mathbf{A}[(k+\Delta)T-\tau]}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.89}$$

Replacing $(k+\Delta)T - \tau$ by β and assuming that $\mathbf{u}(\tau)$ is constant during $kT \leq \tau < (k+\Delta)T$, we get

$$\begin{aligned}\mathbf{x}[(k+\Delta)T] &= e^{\mathbf{A}\Delta T}\mathbf{x}(kT) + \int_0^{\Delta T} e^{\mathbf{A}\beta}d\beta\mathbf{B}\mathbf{u}(kT) \\ &= \mathbf{A}_d(\Delta T)\mathbf{x}(kT) + \mathbf{B}_d(\Delta T)\mathbf{u}(kT)\end{aligned} \tag{3.90}$$

where

$$\mathbf{A}_d(\Delta T) = e^{\mathbf{A}\Delta T} \tag{3.91}$$

and

$$\mathbf{B}_d(\Delta T) = \int_0^{\Delta T} e^{\mathbf{A}\beta} d\beta \mathbf{B} \tag{3.92}$$

Therefore, the matrix $\mathbf{A}_d(\Delta T)$ is obtained by replacing T by ΔT in $\mathbf{A}_d$. Similarly, $\mathbf{B}_d(\Delta T)$ is obtained by replacing T by ΔT in $\mathbf{B}_d$.

3.3.7 Euler's Approximation

Discretization of a continuous-time linear model, as presented in Subsection 3.3.3, by the integral approximation method, gives a desired discrete-time linear model. However, in the case of high-order systems, computation of the state transition matrix is very involved, so that in those cases the matrices $\mathbf{A}_d$ and $\mathbf{B}_d$ are calculated approximately by using some simpler methods. The simplest such a method, known as Euler's approximation, is just one of several methods used for numerical solution of differential equations.

The objective of numerical integration is to find a discrete-time counterpart to a continuous-time model

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{m}(t)$$

in the form of a difference equation. The equation obtained, given by a recursive formula, is then easily solved by a digital computer. The integration of the above equation gives

$$\mathbf{x}(t) = \int_{-\infty}^{t} [\mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{m}(\tau)] d\tau$$

For simplicity, the main idea of the Euler method is explained for a scalar case. Consider the first-order system $\dot{x} = ax + bu$. The integration is analogous to the problem of finding the area, within the imposed integration limits, between the curve defined by $f(t) = ax(t) + bu(t)$ and the time axis. This area is approximately equal to the sum of the rectangles in Figure 3.12.

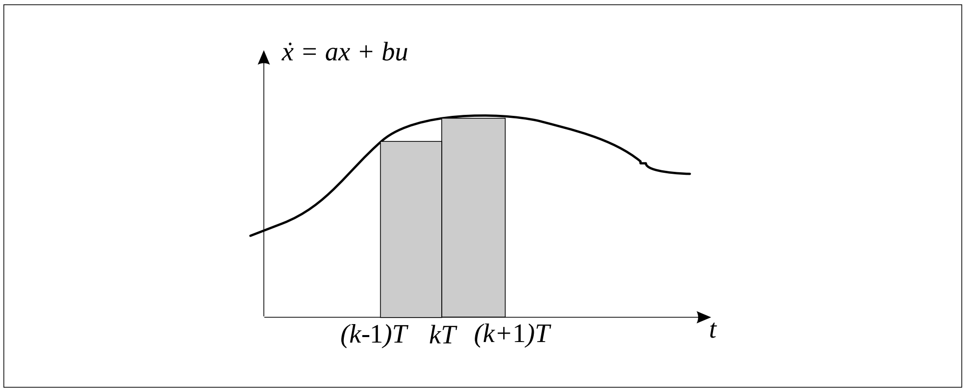

Figure 3.12: Euler's integration method

If the area is calculated according to Figure 3.12, then from the last expression for $t = (k+1)T$ we get

$$\begin{aligned}\mathbf{x}[(k+1)T] &= \int_{-\infty}^{kT} [\mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{m}(\tau)]d\tau + \int_{kT}^{(k+1)T} [\mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{m}(\tau)]d\tau \\ &= \mathbf{x}(kT) + T\mathbf{A}\mathbf{x}(kT) + T\mathbf{B}\mathbf{m}(kT)\end{aligned}$$

or

$$\mathbf{x}[(k+1)T] = (\mathbf{I} + T\mathbf{A})\mathbf{x}(kT) + T\mathbf{B}\mathbf{m}(kT) \tag{3.93}$$

From the last equation, we conclude that for the Euler approximation the state and input matrices are given by

$$\mathbf{A}_d = \mathbf{I} + T \cdot \mathbf{A}, \quad \mathbf{B}_d = T \cdot \mathbf{B} \tag{3.94}$$

It can be observed from (3.70), (3.72), and (3.94) that (3.94) produces only the first two terms of the series expansion given in (3.70) and only the first term of the series expansion given in (3.72). Thus, the Euler approximation is less accurate than the integral approximation considered in Subsection 3.3.3, and for Euler's approximation the sampling interval T must be chosen sufficiently small in order to get satisfactory results.

In general, for more accurate computation of the discrete-time model one can use any known method for numerical solution of differential equations, e.g. the fourth-order Runge–Kutta method or the Adams–Moulton method (Gear, 1971).

3.4 The System Characteristic Equation and Eigenvalues

The characteristic equation is very important in the study of both linear time invariant continuous and discrete systems. No matter what model type is considered (ordinary nth-order differential equation, state space or transfer function), the characteristic equation always has the same form.

If we start with a differential nth-order system model in the operator form

$$\begin{aligned}&\left(p^n + a_{n-1}p^{n-1} + \cdots + a_1 p + a_0\right) y(t)\\ &\quad = \left(b_m p^m + b_{m-1}p^{m-1} + \cdots + b_1 p + b_0\right) u(t)\end{aligned}$$

where the operator p is defined as

$$p^i = \frac{d^i}{dt^i}, \quad i = 1, 2, \ldots, n-1$$

and $m \leq n$, then the *characteristic equation*, according to the mathematical theory of linear differential equations (Boyce and DiPrima, 1992), is defined by

$$s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0 \tag{3.95}$$

Note that the operator p is replaced by the complex variable s playing the role of a derivative in the Laplace transform context.

In the state space variable approach we have seen from (3.54) that

$$\begin{aligned}\mathbf{G}(s) &= \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \frac{1}{|s\mathbf{I} - \mathbf{A}|}\mathbf{C}[\mathrm{adj}(s\mathbf{I} - \mathbf{A})]\mathbf{B} + \mathbf{D}\\ &= \frac{1}{|s\mathbf{I} - \mathbf{A}|}\{\mathbf{C}[\mathrm{adj}(s\mathbf{I} - \mathbf{A})]\mathbf{B} + |s\mathbf{I} - \mathbf{A}|\mathbf{D}\}\end{aligned}$$

The characteristic equation here is defined by

$$|s\mathbf{I} - \mathbf{A}| = 0 \tag{3.96}$$

A third form of the characteristic equation is obtained in the context of the transfer function approach. The transfer function of a single-input single-output system is

$$G(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{3.97}$$

The characteristic equation in this case is obtained by equating the denominator of this expression to zero. Note that for multivariable systems, the characteristic polynomial (obtained from the corresponding characteristic equation) appears in denominators of all entries of the matrix transfer function.

No matter what form of the system model is considered, the characteristic equation is always the same. This is obvious from (3.95) and (3.97), but is not so clear from (3.96). It is left as an exercise to the reader to show that (3.95) and (3.96) are identical (Problem 3.30).

The *eigenvalues* are defined in linear algebra as scalars, λ, satisfying (Fraleigh and Beauregard, 1990)

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{3.98}$$

where the vectors $\mathbf{v} \neq 0$ are called the *eigenvectors*. This system of n linear algebraic equations (λ is fixed) has a solution $\mathbf{v} \neq 0$ if and only if

$$|\lambda\mathbf{I} - \mathbf{A}| = 0 \tag{3.99}$$

Obviously, (3.96) and (3.99) have the same form. Since (3.96) = (3.95), it follows that the last equation is the characteristic equation, and hence the eigenvalues are the zeros of the characteristic equation. For the characteristic equation of order n, the number of eigenvalues is equal to n. Thus, the roots of the characteristic equation in the state space context are the eigenvalues of the matrix $\mathbf{A}$. These roots in the transfer function context are called the *system poles*, according to the mathematical tools for analysis of these systems—the complex variable methods.

Similarity Transformation

We have pointed out before that a system modeled by the state space technique may have many state space forms. Here, we establish a relationship among those state space forms by using a linear transformation known as the similarity transformation.

For a given system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

we can introduce a new state vector $\hat{\mathbf{x}}$ by a linear coordinate transformation as follows

$$\mathbf{x} = \mathbf{P}\hat{\mathbf{x}}$$

where $\mathbf{P}$ is some nonsingular $n \times n$ matrix. A new state space model is obtained as

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \hat{\mathbf{A}}\hat{\mathbf{x}} + \hat{\mathbf{B}}\mathbf{u}, \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_o \\ \mathbf{y} &= \hat{\mathbf{C}}\hat{\mathbf{x}} + \hat{\mathbf{D}}\mathbf{u} \end{aligned} \tag{3.100}$$

where

$$\hat{\mathbf{A}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}, \;\; \hat{\mathbf{B}} = \mathbf{P}^{-1}\mathbf{B}, \;\; \hat{\mathbf{C}} = \mathbf{C}\mathbf{P}, \;\; \hat{\mathbf{D}} = \mathbf{D}, \;\; \hat{\mathbf{x}}(0) = \mathbf{P}^{-1}\mathbf{x}(0) \tag{3.101}$$

This transformation is known in the literature as the *similarity transformation.* It plays an important role in linear control system theory and practice.

Very important features of this transformation are that under similarity transformation both the system eigenvalues and the system transfer function are invariant.

Eigenvalue Invariance

A new state space model obtained by the similarity transformation does not change internal structure of the model, that is, the eigenvalues of the system remain the same. This can be shown as follows

$$\begin{aligned} \left|s\mathbf{I} - \hat{\mathbf{A}}\right| &= \left|s\mathbf{I} - \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right| = \left|\mathbf{P}^{-1}(s\mathbf{I} - \mathbf{A})\mathbf{P}\right| \\ &= \left|\mathbf{P}^{-1}\right||s\mathbf{I} - \mathbf{A}||\mathbf{P}| = |s\mathbf{I} - \mathbf{A}| \end{aligned} \tag{3.102}$$

Note that in this proof the following properties of the matrix determinant have been used

$$\det(\mathbf{M}_1\mathbf{M}_2\mathbf{M}_3) = \det\mathbf{M}_1 \times \det\mathbf{M}_2 \times \det\mathbf{M}_3$$

$$\det\mathbf{M}^{-1} = \frac{1}{\det\mathbf{M}}$$

see Appendix C.

Transfer Function Invariance

Another important feature of the similarity transformation is that the transfer function remains the same for both models, which can be shown as follows

$$\begin{aligned} \hat{\mathbf{G}}(s) = \hat{\mathbf{C}}\left(s\mathbf{I} - \hat{\mathbf{A}}\right)^{-1}\hat{\mathbf{B}} + \hat{\mathbf{D}} &= \mathbf{C}\mathbf{P}\left(s\mathbf{I} - \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right)^{-1}\mathbf{P}^{-1}\mathbf{B} + \mathbf{D} \\ &= \mathbf{C}\mathbf{P}\left[\mathbf{P}^{-1}(\mathbf{sI} - \mathbf{A})\mathbf{P}\right]^{-1}\mathbf{P}^{-1}\mathbf{B} + \mathbf{D} \\ &= \mathbf{C}\mathbf{P}\mathbf{P}^{-1}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}\mathbf{P}^{-1}\mathbf{B} + \mathbf{D} \\ &= \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \mathbf{G}(s) \end{aligned} \tag{3.103}$$

Note that we have used in (3.103) the matrix inversion property (Appendix C)

$$(\mathbf{M}_1\mathbf{M}_2\mathbf{M}_3)^{-1} = \mathbf{M}_3^{-1}\mathbf{M}_2^{-1}\mathbf{M}_1^{-1}$$

The above result is quite logical—the system preserves its input–output behavior no matter how it is mathematically described.

Modal Transformation

One of the most interesting similarity transformations is the one that puts matrix $\mathbf{A}$ into diagonal form. Assume that $\mathbf{P} = \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$, where $\mathbf{v}_i$ are the eigenvectors. We then have

$$\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \hat{\mathbf{A}} = \Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n) \tag{3.104}$$

It is easy to show that the elements λ_i, $i = 1, ..., n$, on the matrix diagonal of Λ are the roots of the characteristic equation $|s\mathbf{I} - \Lambda| = |s\mathbf{I} - \mathbf{A}| = 0$, i.e. they are the eigenvalues. This can be shown in a straightforward way

$$\begin{aligned} |s\mathbf{I} - \Lambda| &= \det\{diag(s - \lambda_1, s - \lambda_2, \ldots, s - \lambda_n)\} \\ &= (s - \lambda_1)(s - \lambda_2)\cdots(s - \lambda_n) \end{aligned}$$

The state transformation (3.104) is known as the *modal transformation*.

Note that the pure diagonal state space form defined in (3.104) can be obtained only in the following three cases.

1. The system matrix has distinct eigenvalues, namely $\lambda_1 \neq \lambda_2 \neq \cdots \neq \lambda_n$.
2. The system matrix is symmetric (see Appendix C).
3. The system minimal polynomial does not contain multiple eigenvalues. For the definition of the minimal polynomial and the corresponding pure diagonal Jordan form, see Subsection 4.2.4.

In the above three cases we say that the system matrix is diagonalizable.

Remark: Relation (3.104) may be represented in another form, that is

$$\mathbf{V}^{-1}\mathbf{A} = \Lambda\mathbf{V}^{-1}$$

or

$$\mathbf{W}^T\mathbf{A} = \Lambda\mathbf{W}^T$$

where

$$\mathbf{W}^T = \mathbf{V}^{-1} \ \Rightarrow \ \mathbf{W}^T\mathbf{V} = \mathbf{I}$$

In this case the *left eigenvectors* $\mathbf{w}_i,\ i = 1, 2, ..., n$, can be computed from

$$\mathbf{w}_i^T \mathbf{A} = \lambda_i \mathbf{w}_i^T \quad \Rightarrow \quad \mathbf{A}^T \mathbf{w}_i = \lambda_i \mathbf{w}_i$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$. Since $|\lambda \mathbf{I} - \mathbf{A}| = \left|\lambda \mathbf{I} - \mathbf{A}^T\right|$, then λ_i is also an eigenvalue of $\mathbf{A}^T$.

There are numerous program packages available to compute both the eigenvalues and eigenvectors of a matrix. In MATLAB this is done by using the function `eig`.

3.4.1 Multiple Eigenvalues

If the matrix $\mathbf{A}$ has multiple eigenvalues, it is possible to transform it into a block diagonal form by using the transformation

$$\mathbf{J} = \mathbf{V}^{-1} \mathbf{A} \mathbf{V} \tag{3.105}$$

where the matrix $\mathbf{V}$ is composed of n linearly independent, so-called *generalized eigenvectors* and $\mathbf{J}$ is known as the Jordan canonical form. This block diagonal form contains simple Jordan blocks on the diagonal. Simple Jordan blocks have the given eigenvalue on the main diagonal, ones above the main diagonal with all other elements equal to zero. For example, a simple Jordan block of order four is given by

$$\mathbf{J}_i(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & 0 & 0 \\ 0 & \lambda_i & 1 & 0 \\ 0 & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & \lambda_i \end{bmatrix}$$

Let the eigenvalue λ_1 have multiplicity of order 3 in addition to two real and distinct eigenvalues, $\lambda_2 \neq \lambda_3$; then a $\mathbf{J}$ matrix of order 5×5 may contain the following three simple Jordan blocks

$$\mathbf{J} = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & 0 & \lambda_3 \end{bmatrix}$$

However, other choices are also possible. For example, we may have the following distribution of simple Jordan blocks

$$\mathbf{J}=\begin{bmatrix}\lambda_1 & 1 & 0 & 0 & 0\\ 0 & \lambda_1 & 0 & 0 & 0\\ 0 & 0 & \lambda_1 & 0 & 0\\ 0 & 0 & 0 & \lambda_2 & 0\\ 0 & 0 & 0 & 0 & \lambda_3\end{bmatrix} \quad \text{or} \quad \mathbf{J}=\begin{bmatrix}\lambda_1 & 0 & 0 & 0 & 0\\ 0 & \lambda_1 & 0 & 0 & 0\\ 0 & 0 & \lambda_1 & 0 & 0\\ 0 & 0 & 0 & \lambda_2 & 0\\ 0 & 0 & 0 & 0 & \lambda_3\end{bmatrix}$$

The study of the Jordan form is quite complex. Much more about the Jordan form will be presented in Chapter 4, where we study system stability.

3.4.2 Modal Decomposition

Diagonalization of matrix $\mathbf{A}$ using transformation $\mathbf{x}=\mathbf{V}\hat{\mathbf{x}}$ makes the system $\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{u}$ diagonal, that is

$$\dot{\hat{\mathbf{x}}}=\Lambda\hat{\mathbf{x}}+\left(\mathbf{V}^{-1}\mathbf{B}\right)\mathbf{u}=\Lambda\hat{\mathbf{x}}+\left(\mathbf{W}^T\mathbf{B}\right)\mathbf{u}, \quad \hat{\mathbf{x}}(0)=\hat{\mathbf{x}}_{\mathrm{o}}$$

In such a case the homogeneous equation $\dot{\mathbf{x}}=\mathbf{A}\mathbf{x},\ \mathbf{x}(0)=\mathbf{x}_{\mathrm{o}}$, becomes

$$\dot{\hat{\mathbf{x}}}=\Lambda\hat{\mathbf{x}}, \quad \hat{\mathbf{x}}(0)=\mathbf{V}^{-1}\mathbf{x}(0)=\mathbf{V}^{-1}\mathbf{x}_{\mathrm{o}}$$

or

$$\dot{\hat{x}}_i=\lambda_i\hat{x}_i, \ \ i=1,\ldots,n$$

This system is represented by n independent differential equations. The modal response to the initial condition is

$$\hat{\mathbf{x}}(t)=e^{\Lambda t}\hat{\mathbf{x}}_{\mathrm{o}}=e^{\Lambda t}\mathbf{V}^{-1}\mathbf{x}_{\mathrm{o}}=e^{\Lambda t}\mathbf{W}^T\mathbf{x}_{\mathrm{o}}$$

or

$$\hat{x}_i(t)=\hat{x}_i(0)e^{\lambda_i t}=\left(\mathbf{w}_i^T\mathbf{x}_{\mathrm{o}}\right)e^{\lambda_i t}$$

The response $\mathbf{x}(t)$ is a combination of the modal components

$$\begin{aligned}\mathbf{x}(t)&=\mathbf{V}\hat{\mathbf{x}}(t)=\mathbf{V}e^{\Lambda t}\mathbf{V}^{-1}\mathbf{x}_{\mathrm{o}}=\mathbf{V}e^{\Lambda t}\mathbf{W}^T\mathbf{x}_{\mathrm{o}}\\ &=\left(\mathbf{w}_1^T\mathbf{x}_{\mathrm{o}}\right)e^{\lambda_1 t}\mathbf{v}_1+\left(\mathbf{w}_2^T\mathbf{x}_{\mathrm{o}}\right)e^{\lambda_2 t}\mathbf{v}_2+\cdots+\left(\mathbf{w}_n^T\mathbf{x}_{\mathrm{o}}\right)e^{\lambda_n t}\mathbf{v}_n\end{aligned} \tag{3.106}$$

This equation represents the modal decomposition of $\mathbf{x}(t)$ and it shows that the total response consists of a sum of responses of all individual modes. Note that $\mathbf{w}_i^T \mathbf{x}_0$ are scalars.

It is customary to call the reciprocals of λ_i the *system time constants* and denote them by τ_i, that is

$$\tau_i = \frac{1}{\lambda_i}, \quad i = 1, 2, ..., n$$

This has physical meaning since the system dynamics is determined by its time constants and these do appear in the system response in the form e^{-t/τ_i}.

The transient response of the system may be influenced differently by different modes, depending of the eigenvalues λ_i. Some modes may decay faster than the others. Some modes might be dominant in the system response. These cases will be illustrated in Chapter 6.

Remark: A similarity transformation $\Lambda = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ can be used for the state transition matrix calculation. Recall

$$\hat{\mathbf{x}}(t) = e^{\Lambda t}\hat{\mathbf{x}}(0), \;\; \hat{\mathbf{x}}(t) = \mathbf{V}\mathbf{x}(t), \;\; \hat{\mathbf{x}}(0) = \mathbf{V}\mathbf{x}(0)$$

and

$$\mathbf{x}(t) = \mathbf{V}^{-1}e^{\Lambda t}\mathbf{V}\mathbf{x}(0) = \Phi(t)\mathbf{x}(0)$$

Hence,

$$\Phi(t) = e^{\Lambda t} = \mathbf{V}^{-1}e^{\Lambda t}\mathbf{V} = \mathbf{W}^T e^{\Lambda t}\mathbf{V}$$

or, in the complex domain

$$\Phi(s) = \mathbf{V}^{-1}(s\mathbf{I} - \Lambda)^{-1}\mathbf{V}$$

$$= \mathbf{V}^{-1}diag\{s - \lambda_1, s - \lambda_2, \ldots, s - \lambda_n\}^{-1}\mathbf{V}$$

$$= \mathbf{V}^{-1}diag\left\{\frac{1}{s - \lambda_1}, \frac{1}{s - \lambda_2}, \ldots, \frac{1}{s - \lambda_n}\right\}\mathbf{V}$$

Remark: The presented theory about the system characteristic equation, eigenvalues, eigenvectors, similarity and modal transformations can be applied directly to discrete-time linear systems with $\mathbf{A}_d$ replacing $\mathbf{A}$.

3.5 State Space MATLAB Laboratory Experiments

In this section we present three MATLAB laboratory experiments on the state space method in control systems. These experiments can be used either as supplements for lectures or independently in the corresponding control system laboratory. Most of the required MATLAB functions have been already introduced in the examples done in this chapter. Students should also consult Appendix D, where a shortened MATLAB manual is given. It is advisable that before using any MATLAB function, the students check all its options by typing `help function name`.

3.5.1 Experiment 1—The Inverted Pendulum

Part 1. The linearized equations of the inverted pendulum, obtained by assuming that the pendulum mass is concentrated at its center of gravity (Kwakernaak and Sivan, 1972; Kamen, 1990) are given by

$$\begin{gathered}\left(J+mL^2\right)\ddot{\theta}(t)-mgL\theta(t)+mL\ddot{d}(t)=0\\(M+m)\ddot{d}(t)+mL\ddot{\theta}(t)=u(t)\end{gathered}\tag{3.107}$$

where $\theta(t)$ is the angle of the pendulum from the vertical position, $d(t)$ is the position of the cart, $u(t)$ is the force applied to the cart, M is the mass of the cart, m is the mass of the pendulum, g is the gravitational constant, and J is the moment of inertia about the center of mass. Assuming that normalized values are given by $J=1$, $L=1$, $g=9.81$, $M=1$, $m=0.1$, derive the state space form

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

where

$$\mathbf{x}(t)=\left[\theta(t)\quad\dot{\theta}(t)\quad d(t)\quad\dot{d}(t)\,\right]^T$$

and $\mathbf{A}^{4\times4}$ and $\mathbf{B}^{4\times1}$ are the corresponding matrices.

Part 2. Using MATLAB determine the following:

(a) The eigenvalues, eigenvectors, and characteristic polynomial of matrix $\mathbf{A}$.

(b) The state transition matrix at the time instant $t=1$.

(c) The unit impulse response (take $\theta(t)$ and $d(t)$ as the output variables) for $0\leq t\leq 1$ with the step size $\Delta t=0.1$ and draw the system response using the MATLAB function `plot`.

(d) The unit step response for $0 \leq t \leq 1$ and $\Delta t = 0.1$. Draw the system response.

(e) The unit ramp response for $0 \leq t \leq 1$ and $\Delta t = 0.1$ and draw the system response. Compare the response diagrams obtained in (c), (d), and (e).

(f) The state response resulting from the initial state $\mathbf{x}(0) = [-1 \quad 1 \quad 1 \quad 1]^T$ and the input $\mathbf{u}(t) = \sin(t)$ for $0 \leq t \leq 5$ and $\Delta t = 0.1$.

(g) The inverse of the state transition matrix $\left(e^{\mathbf{A}t}\right)^{-1}$ for $t = 5$.

(h) The state $\mathbf{x}(t)$ at time $t = 5$ assuming that $\mathbf{x}(10) = [10 \quad 0 \quad 5 \quad 2]^T$ and $u(t) = 0$ by using the result from (g).

(i) Find the system transfer function.

Part 3. Discretize the continuous-time system given in (3.107) with $T = 0.02$ and find the discrete space model

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

Assuming that the output equation of the discrete system is given by

$$\mathbf{y}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(k) = \mathbf{C}_d\mathbf{x}(k)$$

find the system (output) response for $0 \leq k \leq 50$ due to initial conditions $\mathbf{x}_0 = [-1 \quad 1 \quad -1 \quad 1]^T$ and unit step input (note that $u(k)$ should be generated as a column vector of 50 elements equal to 1).

Part 4. Consider the continuous-time system given by

$$\frac{d^2y(t)}{dt^2} + 0.1\frac{dy(t)}{dt} = u(t) \tag{3.108}$$

(a) Discretize this system with $T = 1$ by using the Euler approximation.

(b) Find the response of the obtained discrete system for $k = 1, 2, 3, ..., 20$, when $u(t) = \sin(0.1\pi t)$ and $y(0) = \dot{y}(0) = 0$.

(c) Find discrete transfer function, characteristic equation, eigenvalues, and eigenvectors.

Part 5. Discretize the state space form of (3.108) obtained by using MATLAB function `c2d` with $T = 1$. Find the discrete system response for the initial condition and the input function defined in Part 4b. Compare the results obtained in Parts 4 and 5. Comment on the results obtained.

3.5.2 Experiment 2—Response of Continuous Systems

Part 1. Consider a continuous-time linear system represented by its transfer function

$$G(s) = \frac{s+5}{s^2+5s+6}$$

(a) Find the impulse response by using the MATLAB function `impulse`. In this case you have to use `[y,x]=impulse(num,den)`, where `num` and `den` are row vectors that contain the polynomial coefficients in descending powers of s. Plot both state space variable and output responses (use function `plot`).

(b) Find the step response by using the function `step` and plot both the state response and the output response.

(c) Find the zero-state response due to an input given by $f(t) = e^{-3t}, \quad t \geq 0$. Note that you have to use the function `lsim` and specify input at every time instant of interest. That can be obtained by `t=0:0.1:5` (defines t at $0, 0.1, 0.2, ..., 4.9, 5$) and by `f = exp(-3*t)`. Check that the results obtained in (c) agree with analytical results at $t = 1$.

(d) Obtain the state space form for this system by using the function `tf2ss`. Repeat parts (a), (b), and (c) for the corresponding state space representation. Use the following MATLAB instructions

```
[y,x]=impulse(A,B,C,D,1);
[y,x]=step(A,B,C,D,1);
[y,x]=lsim(A,B,C,D,f,t);
```

respectively, with f and t defined in (c). Compare the results obtained.

Part 2. Consider the continuous-time linear system represented by

$$\frac{d^2y(t)}{dt^2} + 4\frac{dy(t)}{dt} + 4y(t) = \frac{df(t)}{dt} + f(t)$$
$$f(t) = e^{-4t}, \quad t \geq 0, \qquad y\left(0^-\right) = 2, \quad \dot{y}(0) = 1$$

(a) Find the complete system response by using the MATLAB function `lsim`. Compare the simulation results obtained with analytical results. Hint: Use

```
[y,x]=lsim(A,B,C,D,f,t,X0);
```

with t = 0:0.1:5. Note that the initial condition for the state vector, `X0`, has to be found. This can be obtained by playing algebra with the state and output equations and setting $t = 0$.

(b) Find the zeros and poles of this system by using the function `tf2zp`.

(c) Find the system response due to initial conditions specified in Part 2a and the impulse delta function as an input. Since you are not able to specify the system input in time (the delta function has no time structure), you cannot use the `lsim` function. Instead use the `initial` function (zero-input response). The required response is obtained analytically as follows

$$\mathbf{x}(t) = e^{\mathbf{A}t}(\mathbf{x}(0) + \mathbf{B})$$

where $\mathbf{A}$ and $\mathbf{B}$ stand for the system and input matrices in the state space. Thus, the new initial condition is given by $\mathbf{x}(0) + \mathbf{B}$.

(d) Justify the answer obtained in (c). Solve the same problem analytically by using the Laplace transform. Plot results from (c) and compare with results obtained in (d). Can you draw any conclusion for this "nonstandard" problem from the point of view of the system initial conditions at $t = 0^+$. (The standard problem requires that for the impulse response all initial conditions are set to zero.)

Part 3. Given the following dynamical system represented in the state space form by (Gajić and Shen, 1993)

$$\mathbf{A} = \begin{bmatrix} -0.01357 & -32.2 & -46.3 & 0 \\ 0.00012 & 0 & 1.214 & 0 \\ -0.0001212 & 0 & -1.214 & 1 \\ 0.00057 & 0 & -9.1 & -0.6696 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.433 \\ 0.1394 \\ -0.1394 \\ -0.1577 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \mathbf{0}^{2\times 1}$$

This is a real mathematical model of an F-8 aircraft (Teneketzis and Sandell, 1977). Using MATLAB, determine the following quantities.

(a) The eigenvalues, eigenvectors, and characteristic polynomial. Take `p=poly(A)` and verify that `roots(p)` produces also the eigenvalues of matrix $\mathbf{A}$.

(b) The state transition matrix at the time instant $t = 1$. Use the `expm` function.

(c) The unit impulse response and plot output variables. Hint: Use

```
impulse(A,B,C,D);
```

(d) The unit step response and plot the corresponding output variables.

(e) Let the initial system state be $\mathbf{x}(0) = [-1 \quad 1 \quad 0.5 \quad 1]^T$. Find the response due to an input given by $f(t) = \sin(t), \quad 0 < t < 1000$. Hint: Take `t=0:10:1000` and find the corresponding values for $f(t)$ by using the function `sin` in the form f = `sin`(t). Then use the `lsim` function.

(f) Find the system transfer functions. Note that you have one input and two outputs which implies two transfer functions. Hint: Use the function `ss2tf`.

(g) Find the inverse of the state transition matrix $\left(e^{\mathbf{A}t}\right)^{-1} = e^{-\mathbf{A}t}$ at t = 2.

Part 4. Consider a linear continuous-time dynamical system represented by its transfer function

$$G(s) = \frac{(s+1)(s+3)(s+5)(s+7)}{s(s+2)(s+4)(s+6)(s+8)(s+10)}$$

(a) Input the system zeros and poles as column vectors. Note that in this case the static gain k = 1. Use the function `zp2ss(z,p,k)` in order to get the state space matrices.

(b) Find the eigenvalues and eigenvectors of matrix $\mathbf{A}$.

(c) Verify that the transformation $\mathbf{x} = \mathbf{P}\tilde{\mathbf{x}}$, where $\mathbf{P}$ is the matrix whose columns are the eigenvectors of matrix $\mathbf{A}$, produces in the new coordinates the diagonal system matrix $\mathbf{\Lambda} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ with diagonal elements equal to the eigenvalues of matrix $\mathbf{A}$.

(d) Find the remaining state space matrices in the new coordinates. Find the transfer function in the new coordinates and compare it with the original one.

(e) Compare the unit step responses of the original and transformed systems.

3.5.3 Experiment 3—Response of Discrete Systems

Part 1. Consider a discrete-time linear system represented by its transfer function

$$G(z) = \frac{z}{4z^2 + 4z + 1}$$

(a) Find the impulse response by using the MATLAB function `dimpulse`. In this case you have to use `[y,x]=dimpulse(num,den)`, where `num` and `den` are row vectors which contain the polynomial coefficients in descending powers of s. Plot both state and output responses (use function `plot`).

(b) Find the step response by using the function `dstep` and plot both the state and output responses.

(c) Find the system (output) response due to a unit step function, $f(k) = h(k)$, and initial conditions specified by $y(-1) = 0,\ y(-2) = 1$. Note that you have to use the function `dlsim` and to specify input at every time instant of interest. That can be obtained by `k=0:1:20` (defines k at $0, 1, 2, ..., 19, 20$) and by `f(k)=1`. Check analytically that the results obtained in (c) agree with the analytical results for $k = 10$.

(d) Obtain the state space form for this system by using the function `tf2ss`. Repeat parts (a), (b), and (c). Use the following MATLAB statements

```
[y,x]=dimpulse(A,B,C,D,1);
[y,x]=dstep(A,B,C,D,1);
[y,x]=dlsim(A,B,C,D,f,k);
```

respectively, with f and k defined in (c). Compare the results obtained.

Part 2. Consider the discrete-time linear system represented by

$$y(k+2) + \frac{5}{6}y(k+1) + \frac{1}{6}y(k) = f(k+1)$$
$$f(k) = (0.8)^k u(k), \qquad y(-1) = 2, \quad y(-2) = 3$$

(a) Find the system response by using the MATLAB function `dlsim`. Hint: Use

`[y,x]=dlsim(A,B,C,D,f,X0);` with `k=0:1:10`.

Note that the initial condition has to be found. This can be obtained by playing algebra with the state space and output equations. Compare the simulation results obtained with analytical results.

(b) Find the zeros and poles of this system by using the function `tf2zp`.

(c) Find the system response due to initial conditions specified in Part 2a and with the impulse delta function as an input. Use the `dlsim` function.

(d) Solve the same problem analytically by using the $\mathcal{Z}$-transform. Plot results from (c) and compare with results obtained in (d).

Part 3. Given a dynamical system represented in the continuous-time state space form in Section 3.5.2, Experiment 2, Part 3.

(a) Discretize the continuous-time system by using the MATLAB function `c2d`. Assume that the sampling period is $T = 1$.

(b) Find the eigenvalues, eigenvectors, and characteristic polynomial of the obtained discrete-time system.

(c) Find the state transition matrix at time instant $k = 5$.

(d) Find the unit impulse response and plot output variables. Hint: Use

```
dimpulse(A,B,C,D);
```

(e) Find the unit step response and plot the corresponding output variables.

(f) Assume that the initial system state is $\mathbf{x}(0) = [-1 \quad 0 \quad 1 \quad -0.5]^T$. Find the response due to an input given by $f(t) = sink, \ \ 0 < k < 1000$. Hint: Take `k=0:10:1000` and find the corresponding values for $f(k)$ by using the function `sin` in the form f = `sin`(k). Then use the `dlsim` function. Compare the obtained discrete-time results with the continuous-time results for the same system studied in Section 3.5.2, Experiment 2.

(g) Find the system transfer functions. Note that you have one input and two outputs which implies two transfer functions. The matrices $\mathbf{C}$ and $\mathbf{D}$ are not changed due to discretization procedure. Hint: Use the function `ss2tf`.

3.6 References

Arkun, Y. and S. Ramakrishnan, "Bounds of the optimum quadratic cost of structure constrained regulators," *IEEE Transactions on Automatic Control*, vol. AC-28, 924–927, 1983.

Boyce, W. and R. DiPrima, *Elementary Differential Equations*, Wiley, New York, 1992.

Chen, C., *Linear System Theory Design*, CBS College Publishing, New York, 1984.

Dressler, R. and D. Tabak, "Satellite tracking by combined optimal estimation and control techniques," *IEEE Transactions on Automatic Control*, vol. AC-16, 833–840, 1971.

Fraleigh, J. and R. Beauregard, *Linear Algebra*, Addison-Wesley, Reading, Massachusetts, 1990.

Gajić, Z. and X. Shen, "Study of the discrete singularly perturbed linear quadratic control problem by a bilinear transformation," *Automatica*, vol. 27, 1025–1028, 1991.

Gajić, Z. and X. Shen, *Parallel Algorithms for Optimal Control of Large Scale Linear Systems*, Springer-Verlag, London, 1993.

Gear, W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice Hall, Englewood Cliffs, New Jersey, 1971.

Grodt, T. and Z. Gajić, "The recursive reduced order numerical solution of the singularly perturbed differential Riccati equation," *IEEE Transactions on Automatic Control*, vol. AC-33, 751–754, 1988.

Kailath, T., *Linear Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.

Kamen, E., *Introduction to Signals and Systems*, Macmillan, New York, 1990.

Kokotović, P., J. Allemong, J. Winkelman, and J. Chow, "Singular perturbation and iterative separation of the time scales," *Automatica*, vol. 16, 23–33, 1980.

Kwakernaak, H. and R. Sivan, *Linear Optimal Control Systems*, Wiley, New York, 1972.

Litkouhi, B., *Sampled-Data Control Systems with Slow and Fast Modes*, Ph.D. Dissertation, Michigan State University, 1983.

Petkovski, D., N. Harkara, and Z. Gajić, "Fast suboptimal solution to the static output control problem of linear singularly perturbed systems," *Automatica*, vol. 27, 721–724, 1991.

Teneketzis, D. and N. Sandell, "Linear regulator design for stochastic systems by multiple time-scale method," *IEEE Transactions on Automatic Control*, vol. AC-22, 615–621, 1977.

Spong, M., "The swing up control problem for the acrobot," *IEEE Control Systems Magazine*, vol. 15, 49–55, 1995.

3.7 Problems

3.1 An antenna control problem (Dressler and Tabak, 1971) is represented by the open-loop transfer function

$$G(s) = \frac{K(s+1)}{s^2(s+6)(s+11.5)(s^2+8s+256)}$$

Find state space matrices for the following forms:

(a) Controller canonical form.

(b) Observer canonical form.

3.2 A robotic manipulator called the acrobot has the following linearized model (Spong, 1995)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 12.49 & -12.54 & 0 & 0 \\ -14.49 & 29.36 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -2.98 \\ 5.98 \end{bmatrix}$$

Assume that the output matrices are given by

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

Use MATLAB in order to find the following quantities:

(a) Eigenvalues and characteristic polynomial.
(b) Modal canonical form.
(c) Open-loop transfer function.
(d) Controller and observer canonical forms.

3.3 Consider the harmonic oscillator in the state space form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \qquad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$y = [0 \quad 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(a) Find the state transition matrix.
(b) Find the system response due to a unit step input.
(c) Verify the answer obtained by using the MATLAB functions `ss2zp` and `lsim`.

3.4 Given the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}$$

Find $e^{\mathbf{A}t}$ by the Cayley–Hamilton and the Laplace transform methods.

3.5 For the system

$$\ddot{y}(t) + 2\dot{y}(t) + y(t) = 6\dot{u}(t) + u(t)$$

(a) Draw the simulation block diagram. Use any method.
(b) Find the system impulse response by the MATLAB function `impulse`.

3.6 Given a discrete system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$$

Find its response due to the initial condition given by $\mathbf{x}(0) = [1 \quad 1]^T$.

3.7 Given a linear time invariant continuous system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

with

$$\mathbf{x}(1) = \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \quad u(t) = \begin{cases} 1 & t \geq 2 \\ 0 & 0 < t \leq 2 \end{cases}$$

Assuming that the state transition matrix has a known (given) form as

$$\Phi(t - t_0) = \begin{bmatrix} \phi_{11}(t - t_0) & 0 \\ \phi_{21}(t - t_0) & \phi_{22}(t - t_0) \end{bmatrix}$$

find the system response for any $t \geq 0$.

3.8 Find the impulse response of the system

$$\ddot{y} + 2\dot{y} + 10y = \ddot{u} - 3\dot{u} + 5u$$

Find the system transfer function and the state space form.

3.9 Find the system response for $t > 1$ due to its initial condition at $t = 1$

$$\frac{dy(t)}{dt} + 4y(t) + 3\int y(\tau)d\tau = 0, \qquad y(1) = 2$$

3.10 Find the response of the discrete system

$$y(k+2) + y(k) = (-1)^k, \quad y(0) = 1, \; y(1) = 0$$

Verify the answer by the MATLAB function `dlsim`.

3.11 A continuous-time system is represented by

$$\ddot{y} + 4\dot{y} + 3y = u$$

(a) Find the transfer function and the impulse response.
(b) Compute the response $y(t)$ for $y(0^-) = -2$, $\dot{y}(0^-) = 1$ and $u(t)$ equal to a unit step function.

3.12 Given a linear continuous-time system with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},\ \mathbf{x}(0) = \begin{bmatrix} -1 \\ 0 \end{bmatrix},\ \mathbf{C} = \mathbf{I},\ \mathbf{D} = \mathbf{0}$$

(a) Find the state transition matrix.
(b) Find the system transfer function.
(c) Find the system response due to a unit step input.
(d) Verify the answers obtained by using MATLAB.

3.13 A discrete system is given by

$$y(k+1) - 0.5y(k) = 2u(k+1) + u(k)$$

Compute the impulse response. Verify the result by the MATLAB function `dimpulse`.

3.14 Discretize the following system by using the Euler approximation

$$\ddot{y} + 2\dot{y} + 3\sin(y(t)) = u(t), \quad y(0) = 1,\ \dot{y}(0) = 2$$

3.15 Given a time invariant linear system with the impulse response equal to e^{-t}. Find the response of this system due to an input $2\delta(t-1) + 3h(t-2)$, where $\delta(t)$ is the impulse delta function and $h(t)$ is a unit step function. What MATLAB functions can be used to solve this problem?

3.16 A linear discrete system is represented by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ \mathbf{x}(0) = \begin{bmatrix} -1 \\ 0 \end{bmatrix},\ \mathbf{C} = [0 \quad 1],\ \mathbf{D} = 0$$

(a) Find its state transition matrix.
(b) Find the transfer function.
(c) Find the system response due to $u(k) = k$ assuming that $x_1(0) = 1$ and $x_2(0) = 3$.

3.17 Find the response of the system given by

$$\ddot{y} + 2\dot{y} + y = \dot{u} + u, \quad u(t) = 2e^{-t}, \quad y(0) = 1,\ \dot{y}(0) = 1$$

3.18 Given a second-order linear system at rest (initial conditions are zero)

$$\ddot{y} + 2\xi\omega_n\dot{y} + \omega_n^2 y = \omega_n^2 u(t)$$

Find its unit step response for $\xi < 1$.

3.19 Find the response of the discrete system

$$y(k+2) - 6y(k+1) + 8y(k) = 3k + 2, \quad y(0) = 1,\ y(1) = 1$$

3.20 Find the response of the continuous system

$$\ddot{y} + 3\dot{y} - 10y = 2\dot{u} + 5u, \quad y(0) = 1,\ \dot{y}(0) = -1, \quad u(t) = t$$

3.21 Discretize the system $\dot{y} = u$ by using both the Euler and the integral approximations. Compare the discrete systems obtained.

3.22 Given a linear continuous system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$$

with

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$$

Find the similarity transformation such that this system has the diagonal form in the new coordinates.

3.23 Find the state transition matrix of a continuous system with

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Use the Taylor series expansion method.

3.24 Find the response of a discrete system represented by

$$y(k+2) + 2y(k) + 1 = (-1)^k, \quad y(0) = y(1) = 1$$

3.25 Find the transition matrix in the complex domain for the system represented by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

3.26 Consider a fifth-order industrial reactor (Arkun and Ramakrishnan, 1983; Petkovski *et al.*, 1991) represented by

$$\mathbf{A}=\begin{bmatrix}-16.11 & -0.39 & 27.2 & 0 & 0\\ 0.01 & -16.99 & 0 & 0 & 12.47\\ 15.11 & 0 & -53.6 & -16.57 & 71.78\\ -53.36 & 0 & 0 & -107.2 & 232.11\\ 2.27 & 60.1 & 0 & 2.273 & -102.99\end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix}11.12 & -2.61 & -21.91 & -53.5 & 69.1\\ -12.6 & 3.36 & 0 & 0 & 0\end{bmatrix}^T$$

$$\mathbf{C}=\begin{bmatrix}0 & 0 & 0 & 0 & 1\\ 0 & 1 & 1 & 0 & 0\end{bmatrix}$$

Using MATLAB, find the following:

(a) The system transfer function.
(b) The impulse response.
(c) The response due to inputs $u_1(t)=e^{-t}+\sin(t)$ and $u_2(t)=0$.

3.27 Discretize the system given in Problem 3.26 by the MATLAB function `c2d` with $T=0.1$ and repeat the steps (a), (b), and (c) from Problem 3.26.

3.28 The model of a synchronous machine connected to an infinite bus (Kokotović *et al.*, 1980; Grodt and Gajić, 1988) has the system matrix

$$\mathbf{A}=\begin{bmatrix}-0.58 & 0 & 0 & -0.27 & 0 & 0.2 & 0\\ 0 & -1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & -5 & 2.1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 337 & 0 & 0\\ -0.14 & 0 & 0.14 & -0.2 & -0.28 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.08 & 2\\ -17.2 & 66.7 & -11.6 & 40.9 & 0 & -66.7 & -16.7\end{bmatrix}$$

(a) Find the eigenvalues, eigenvectors, and similarity transformation that puts this system into a diagonal form.
(b) Discretize this system with $T=1$.
(c) Find the response of the discrete system obtained in part (b) due to the initial condition $\mathbf{x}(0)=[1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1]^T$ and draw the corresponding response for the time interval $0\le k\le 10$. Use MATLAB.

3.29 A linearized mathematical model of an aircraft considered in Litkouhi (1983) and Gajić and Shen (1991) has the form

$$\mathbf{A} = \begin{bmatrix} -0.015 & -0.0805 & -0.0011666 & 0 \\ 0 & 0 & 0 & 0.03333 \\ -2.28 & 0 & -0.84 & 1 \\ 0.6 & 0 & -4.8 & -0.49 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -0.0000916 & 0.0007416 \\ 0 & 0 \\ -0.11 & 0 \\ -8.7 & 0 \end{bmatrix}$$

Obtain the following (using MATLAB):

(a) Discretize this model with $T = 1$.
(b) Find its response due to a unit ramp input.
(c) Find the system transfer function and the system poles.

3.30 Show by induction that the characteristic equation (3.96) of a system in the phase variable canonical form is indeed given by (3.95).

3.31 Write a MATLAB program to obtain the system's modal form for Example 3.4. Using that program, check the corresponding results in Examples 3.4 and 3.9.