1 Introduction

An understanding of dynamical system models is important in diverse disciplines: automatic control, communications, filter design, specialized computer circuit design, power systems, and robotics, as well as in other branches of engineering, and indeed, in many branches of the physical, biological and social sciences. Many of the dynamical system models of interest can be put into a common framework, that of the so-called "state-space" model. Engineering examples will be emphasized here, and the important class of linear, constant models will be analyzed in detail.

State-space models are sets of equations of a particular form, with several useful attributes: first, their form contributes to an intuitive understanding of the behavior of many dynamical systems; second, efficient computational techniques are available for solving them; and third, a large body of theory is available for analyzing them.

A general description of state-space equations will be given, and then several simple examples will be put into state-space form. The modeling consequences of linearity and time-invariance will be given, and then the simplest systems, those that are both linear and time-invariant, will be introduced. Finally, a technique for finding a linear approximation of a nonlinear system near a known solution will be given.

1 The structure of state-space models

State-space models are collections of equations corresponding to the logical structure illustrated by Figure 1.1. As will be seen, this logical structure applies to a variety of physical objects, such as electromechanical systems, digital computers, and digital software processes. Writing the equations is extremely simple for a model corresponding to this diagram, but the starting point may be a model of different structure, in which case the corresponding equations have to be manipulated to rewrite them in state-space form.

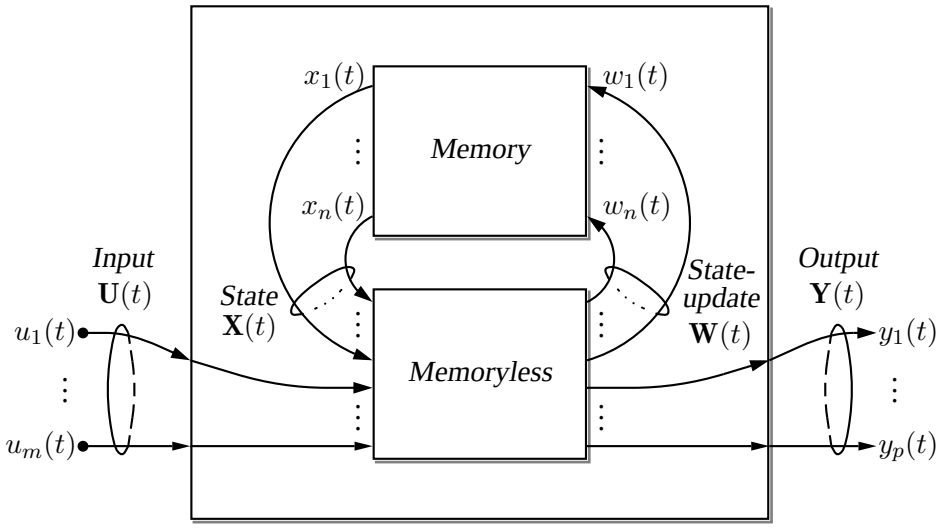

Fig. 1.1 The components of a dynamical system.

All variables in the model are assumed to be a function of an independent variable, usually time, which justifies the adjective "dynamic" for the system. Thus there is a set, typically $\mathbb{T} = \mathbb{R}$, the set of real numbers for analog systems, for which time is continuous, or $\mathbb{T} = \mathbb{Z}$, the integers, for discrete-time systems such as computer circuits.

At each time $t \in \mathbb{T}$, a set of m independent external quantities u_i, $i = 1 \cdots m$ called *inputs* is assumed to affect the system, and at each time the system is assumed to produce a set of p quantities y_i, $i = 1 \cdots p$ called *outputs*, which may affect the external environment.

For modeling purposes, inside the system there are two logically distinct types of components, grouped together in Figure 1.1. There is a set of memoryless components that compute instantaneous functions of the external inputs and of internal variables x_i, $i = 1 \cdots n$ which are called *state* variables. The second set of components contains memory elements that store the internal variables. Thus, the memoryless part computes the functions

$$\begin{aligned} y_1(t) &= g_1(x_1(t), \cdots x_n(t), u_1(t), \cdots u_m(t), t) \\ &\vdots \\ y_p(t) &= g_p(x_1(t), \cdots x_n(t), u_1(t), \cdots u_m(t), t). \end{aligned} \tag{1.1}$$

These functions specify the p system outputs at each time t in terms of the input and state variables at time t, and in terms of t itself. By convention, the above

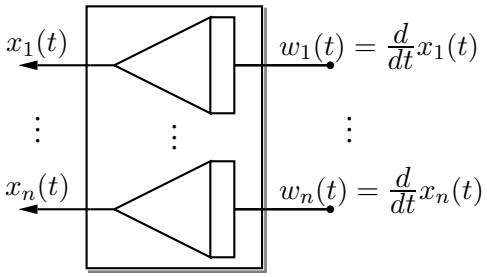

Fig. 1.2 Continuous-time memory.

equations are often written as

$$\mathbf{Y}(t) = \mathbf{G}(\mathbf{X}(t), \mathbf{U}(t), t). \tag{1.2}$$

In this equation, $\mathbf{Y}(t)$ is a column of variables $y_1(t), \cdots y_p(t)$, and is referred to as a vector of dimension p, or a p-vector, with entries (or elements) $y_i(t)$, $i = 1, \cdots p$. Similarly $\mathbf{X}(t)$ is an n-vector and $\mathbf{U}(t)$ is an m-vector. The entries of vector $\mathbf{G}$ are the quantities $g_i(\cdots)$, $i = 1, \cdots p$, which are functions of the entries $x_i(t)$, $i = 1, \cdots n$ of the state vector $\mathbf{X}(t)$; of the entries $u_i(t)$, $i = 1, \cdots m$ of the input vector $\mathbf{U}(t)$; and of t.

Similarly, the state-update values in Figure 1.1 are given by equations of the form

$$\begin{aligned} w_1(t) &= f_1(x_1(t), \cdots x_n(t), u_1(t), \cdots u_m(t), t) \\ &\vdots \\ w_n(t) &= f_n(x_1(t), \cdots x_n(t), u_1(t), \cdots u_m(t), t) \end{aligned} \tag{1.3}$$

and these equations are abbreviated, by convention, as

$$\mathbf{W}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t), t). \tag{1.4}$$

Two possible memory types will be considered. The continuous-time memory shown in Figure 1.2 contains n integrators, for which, using vector notation,

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{W}(t), \tag{1.5}$$

where the time-derivative of a vector is the vector of derivatives of its entries. The discrete-time memory of Figure 1.3 contains n delays, for which

$$\mathbf{X}(t+1) = \mathbf{W}(t). \tag{1.6}$$

State-space forms

When the equations for the memory and memoryless parts are combined to eliminate $\mathbf{W}(t)$, the continuous-time state-space equations take the form

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t), t) \tag{1.7a}$$

$$\mathbf{Y}(t) = \mathbf{G}(\mathbf{X}(t), \mathbf{U}(t), t), \tag{1.7b}$$

and the equations for a discrete-time system have the form

$$\mathbf{X}(t{+}1) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t), t) \tag{1.8a}$$

$$\mathbf{Y}(t) = \mathbf{G}(\mathbf{X}(t), \mathbf{U}(t), t). \tag{1.8b}$$

The right-hand sides of (1.7) and (1.8) contain only state variables and system inputs, and the equations explicitly determine $\frac{d}{dt}\mathbf{X}(t)$ in (1.7), $\mathbf{X}(t{+}1)$ in (1.8), and $\mathbf{Y}(t)$ in both cases. Writing state-space equations from some other starting point requires the above left-hand variables to be solved exclusively in terms of the above right-hand variables.

Often the model inputs and outputs represent real-valued quantities, that is, $\mathbf{U}(t) \in \mathbb{R}^m$ and $\mathbf{Y}(t) \in \mathbb{R}^p$, in which case the state vector $\mathbf{X}(t)$ normally contains real values as well, that is, $\mathbf{X}(t) \in \mathbb{R}^n$; but as will be seen, it may be convenient to allow $\mathbf{X}(t) \in \mathbb{C}^n$, the complex n-vectors.

Other kinds of variables are possible, however. For example, in binary computer circuits, all variables except t take values from the set $\{0, 1\}$, and provided the model is linear, it is possible to write $\mathbf{U}(t) \in \mathbb{Z}_2^m$, $\mathbf{Y}(t) \in \mathbb{Z}_2^p$, $\mathbf{X}(t) \in \mathbb{Z}_2^n$, where $\mathbb{Z}_2$ is the set of integers modulo 2.

In summary, defining a state-space model requires definition of the input, output, and state vectors; the time set $\mathbb{T}$; and the vector functions $\mathbf{F}(\cdots)$, $\mathbf{G}(\cdots)$ in (1.7) or (1.8). Systems described by ordinary differential equations (1.7a) contain integrators in the memory of the corresponding conceptual model. Discrete-time systems described by (1.8a) contain delay elements that store quantities for one time interval. In both circumstances, the state variables are conveniently chosen as the contents of the memory elements.

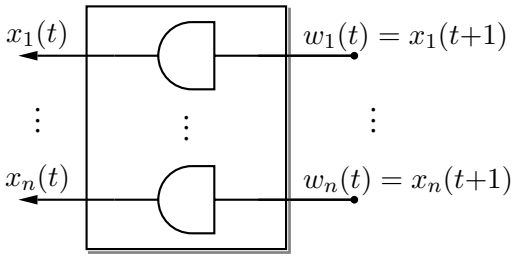

Fig. 1.3 Discrete-time memory.