Chapter 2

Nonlinear Process Identification

RONALD K. PEARSON
BABATUNDE A. OGUNNAIKE

DuPont Central Science and Engineering
Experimental Station,
Wilmington, Delaware, 19880-0101

2.1 Introduction

Most, if not all, of the nonlinear process control strategies described in this book require an explicit mathematical model of process dynamics. This chapter is concerned with the development of models suitable for use in these control system design methodologies. Many of the key issues in the discussions that follow — motivation, methodology, open questions — may be related to the "process characterization cube" shown in Fig. 2.1.

The three axes of this cube correspond to the following three process characteristics:

1. degree of dynamic complexity;

2. degree of nonlinearity;

3. degree of interaction.

Figure 2.1: The Process Characterization Cube

This construct was originally proposed in reference [64], but a more detailed discussion, illustrated with specific process examples, is given in Chapter 30 of reference [66]. Our motivation for developing empirical process models becomes clear when we consider each of the following topics in terms of this cube:

1. real-world process dynamics;
2. fundamental models of process dynamics;
3. empirical models of process dynamics;
4. control-system design methodologies.

Specifically, the design of a process control system requires us to match the first item on this list with the last. In the development of model-based control system designs, this match is to be accomplished through a mathematical model that satisfies two important criteria. First, it must be compatible in structure and complexity with the requirements of the control system design methodology under consideration. As a specific example, the nonlinear Model Predictive Control (MPC) algorithms discussed by Rawlings [80] require discrete-time dynamic models and the computational complexity of

the problem grows rapidly as the model complexity increases. The second criterion for a "good" process model is that it approximate the true process dynamics well enough that the resulting control system will perform adequately in practice.

In general, fundamental models — derived from known conservation laws, reaction kinetics, etc. — can be expected to describe process dynamics more completely than empirical models can. Conversely, since these models are fundamental descriptions of the process, we have limited control over their complexity, which can be substantial. For example, the detailed model given in [21] for the co-polymerization of methyl methacrylate and vinyl acetate entails approximately 50 differential and algebraic equations, mostly nonlinear, and about as many parameters. The advantage of empirical models is that we can specify model complexity explicitly, giving us at least partial control over the difficulty of the resulting control problem. On the other hand, the price we pay for this reduced model complexity is generally reduced model fidelity. Thus, to apply model-based control system design procedures effectively, it is necessary to develop empirical process models that are "close enough" in the process characterization cube to the dynamics of the real-world process, but that are also within the "domain" of the control system design methodology of interest. *The principal objective of this chapter is to describe the key issues that we must confront when we attempt to solve this problem in practice.*

Box and Jenkins proposed a systematic approach to linear time-series modeling, described in reference [10] and consisting of the following four steps:

1. selection of a general class $\mathcal{C}$ of empirical models for consideration;
2. identification of a specific subclass of models to be fit;
3. estimation of model parameters;
4. assessment of model adequacy.

The models considered by Box and Jenkins were *parametric* models, described by a finite number of adjustable parameters (see Section 5.1 for a brief discussion of parametric modeling and its alternatives). Consequently, step 2 of their procedure is primarily concerned with deciding how many model parameters to admit (i.e., how complex to allow the model to become). One of the key points of this chapter (Section 3.2) is that certain equivalences that hold between different linear models (e.g., discrete- and

continuous-time models) *do not hold* for nonlinear models. Consequently, issues of model structure selection — corresponding to steps 1 and 2 of the Box-Jenkins procedure — are even more important in nonlinear modeling than in linear modeling. It is for this reason that we concentrate more on these issues in this chapter than on the "more traditional" topic of parameter estimation algorithms.

The remainder of this chapter is divided into sections, numbered 2 through 10. Sections 2 and 3 deal with continuous-time and discrete-time process models, respectively, providing useful background material for the discussions that follow. Sections 4 and 5 then discuss the empirical model building process and the problem of model structure selection, respectively, both closely related to the first step in the Box-Jenkins model development procedure. Because the subsequent three steps in this procedure have strong connections to statistics, Section 6 gives a brief overview of some useful statistical background material. Section 7 then discusses the problem of model parameter estimation, while Section 8 discusses the problems of cross-validation and input sequence design. Section 9 then illustrates some of the main ideas developed here with a case study (control of a high-purity distillation process) and Section 10 gives a brief summary of these ideas.

2.2 Continuous-Time Process Models

Fundamental process descriptions usually take the form of continuous-time models, so this section gives a brief overview of some of the most popular continuous-time model structures that appear in the control literature. All of these models are special cases of — or closely related to — the class of state-space models:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \mathbf{G}(\mathbf{x}(t)), \end{aligned} \tag{2.1}$$

Here, $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$ are state, input, and output vectors of dimensions n, m, and r, respectively. The vector function $\mathbf{F}$ maps the domain $R^n \times R^m$ into the range R^n, while the vector function $\mathbf{G}$ maps R^n into R^r.

The principal objective of this overview is to give some insight into the range of qualitative behavior these models can exhibit. This insight is important because in empirical modeling, we are attempting to develop a "moderate complexity" model that captures the *essential qualitative behavior* of the real-world process of interest.

2.2.1 Control-Affine Models

Kantor notes [47] that many first-principles models may be represented as *control-affine* special cases of Eq. (2.1). Here, the nonlinear vector map $\mathbf{F}(\mathbf{x}, \mathbf{u})$ has the special form:

$$\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) + \mathbf{\Gamma}(\mathbf{x})\mathbf{u}. \tag{2.2}$$

This structural restriction on $\mathbf{F}(\mathbf{x}, \mathbf{u})$ necessarily limits the class of dynamic phenomena that may be represented, relative to the unconstrained state-space model (2.1). Despite these limitations, however, the discussion given here illustrates that this class of dynamic phenomena is still very broad. As a specific example, the second-order polynomial model considered in Section 4.2 is control-affine, but it is particularly interesting because it can exhibit a form of instability known as *finite escape time.* More generally, note that all linear models are included in the control-affine class, as are the bilinear models discussed in Section 2.2. In addition, Boyd and Chua [12] have shown that the *finite* Volterra models considered in Section 2.3 have bilinear realizations, implying that they, too, are equivalent to members of the class of control-affine models. Before proceeding to more detailed discussions of these important subsets of the control-affine model class, it is useful to briefly consider the following two examples. These examples are not members of the more restricted model classes discussed in Sections 2.2 and 2.3, and they illustrate the range of behavior that the control-affine class of dynamic models — but not all of its sub-classes — is capable of exhibiting.

Example 1 — Chaotic Dynamics

Chaos is a nonlinear dynamic phenomenon that has recieved much attention recently [26, 39, 96]. One of the simplest continuous-time models that is flexible enough to exhibit chaotic dynamics is the Lorenz equation [26]:

$$\begin{aligned} dx/dt &= -\sigma x + \sigma y \\ dy/dt &= -xz + rx - y \\ dz/dt &= xy - bz. \end{aligned} \tag{2.3}$$

Physically, these equations describe the convection-driven patterns that arise in fluid flow between rectangular plates. In this problem, the variables x, y, and z represent the intensity of convective motion, the temperature difference between ascending and descending currents in the fluid, and the deviation from linearity of the vertical temperature profile between the plates.

To see the connection between these equations and the control-affine model (2.2), define the state vector components $x_1 = x$, $x_2 = y$, and $x_3 = z$ and take r as the scalar control input u. In particular, note that r is a ratio of Rayleigh numbers and is proportional to the temperature difference between plates. Thus, if we consider the problem of controlling fluid motion by manipulating this temperature difference, the Lorenz equations may be re-written as a control-affine model by defining:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} -\sigma x_1 + \sigma x_2 \\ -x_1 x_3 - x_2 \\ x_1 x_2 - b x_3 \end{bmatrix}, \tag{2.4}$$

and

$$\mathbf{\Gamma}(\mathbf{x}) = \begin{bmatrix} 0 \\ x_2 \\ 0 \end{bmatrix}. \tag{2.5}$$

This example demonstrates that the class of control-affine models is "qualitatively rich enough" to exhibit chaotic dynamics.

Example 2 — Jump Phenomena

Jump phenomena are important in the theory of nonlinear oscillators [13, 62, 96]. While these phenomena will not be discussed in detail here, they have been widely studied and it is useful to note that the class of control-affine models can also exhibit this type of behavior. This point is significant since some of the interesting subsets of this model class lack this flexibility, as noted in subsequent discussions. Here, note that one of the models often used to illustrate jump phenomena is Duffing's equation:

$$\frac{d^2y}{dt^2} + c\frac{dy}{dt} + \alpha y + \beta y^3 = u(t). \tag{2.6}$$

This model may be cast in control-affine form by defining the state variables $x_1 = y$ and $x_2 = dy/dt$. The functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{\Gamma}(\mathbf{x})$ appearing in Eq. (2.1) are then given by:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_2 \\ -\alpha x_1 - \beta x_1^3 - c x_2 \end{bmatrix}, \tag{2.7}$$

and

$$\mathbf{\Gamma}(\mathbf{x}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.8}$$

2.2.2 Bilinear Models

For single-input problems (i.e., $\mathbf{u}(t) = u(t)$ is a scalar), if the control-affine model is further restricted to $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and $\mathbf{\Gamma}(\mathbf{x}) = \mathbf{N}\mathbf{x} + \mathbf{b}$, the result is a *bilinear* model. Here, $\mathbf{A}$ and $\mathbf{N}$ are $n \times n$ matrices and $\mathbf{b}$ is an n-vector. This class may be extended to MIMO problems by considering multiple terms of this general form, i.e.:

$$\mathbf{\Gamma}(\mathbf{x})\mathbf{u} = \sum_{i=1}^{m} u_i(t)\mathbf{N}_i\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{2.9}$$

where $\mathbf{B}$ is an $n \times m$ matrix. Further, note that when $\mathbf{N}_i = \mathbf{0}$, for all i, the bilinear model reduces to the more familiar linear one. Because this structure is a special case of the more general control-affine structure, we can expect the class of behavior it is capable of representing to be correspondingly restricted. For example, Brockett has shown [13] that bilinear models cannot represent jump phenomena like that exhibited by the Duffing equation.

A point that will be emphasized repeatedly throughout this chapter is that the qualitative behavior of a nonlinear model can depend strongly on the specific inputs considered. The step response of a single-input, single-output bilinear model illustrates this point nicely. Specifically, consider the effect of the step input:

$$u(t) = \begin{cases} \alpha & t > 0 \\ 0 & t \leq 0 \end{cases} \tag{2.10}$$

on the bilinear system defined above. If $\mathbf{x}(t) = \mathbf{0}$ for $t \leq 0$, the bilinear model may be re-written as the "equivalent" linear model for $t > 0$:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (\mathbf{A} + \alpha\mathbf{N})\mathbf{x}(t) + \mathbf{b}u(t). \\ y(t) &= \mathbf{c}^T\mathbf{x}(t) \end{aligned} \tag{2.11}$$

The response of this model will be determined by the eigenvalues of the matrix $\mathbf{A} + \alpha\mathbf{N}$, which depend on the input amplitude α. For low-amplitude inputs (small α) and "weak" nonlinearities ("small" $\mathbf{N}$), these eigenvalues

will be approximately the same as those of **A**, so the effects of the nonlinearity will not be felt. As α increases, however, the qualitative behavior of the step response will come ultimately to be dominated by the eigenvalues of **N**, which may be very different from those of **A**.

2.2.3 Volterra Models and Fading Memory Systems

Another class of nonlinear models that has attracted significant interest is the class of Volterra models [12, 79, 88, 96]. For the single-input, single-output (SISO) case, this series is given by:

$$y(t) = y_0(t) + \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} k_n(t; t_1, t_2, ..., t_n) u(t_1) \cdots u(t_n) dt_1 ... dt_n. \tag{2.12}$$

Here, the functions $k_n(t; t_1, t_2, ..., t_n)$ are called the *Volterra kernels* and generalize the impulse response characterizing a linear system. Essentially, this series represents the response $y(t)$ in terms of the past history of the input $u(\tau)$ for all $\tau < t$ and the kernels must be constrained to enforce this *causality condition.* As a specific example, note that the standard linear convolution model may be represented as a Volterra model with all terms identically zero except the first-order kernel:

$$k_1(t; t_1) = \begin{cases} h(t - t_1) & t_1 < t \\ 0 & t_1 \geq t \end{cases} \tag{2.13}$$

To be useful in practice, Volterra models must be truncated to a finite number of terms. Boyd and Chua [12] consider the question of what class of system behavior may be approximated by such truncated Volterra models. They show that systems with *fading memory* on a particular set K of input sequences may be approximated arbitrarily well on that set by a truncated Volterra model of sufficiently high order. Roughly speaking, a fading memory system is one whose dependence on past inputs decreases "rapidly enough" with time. This concept is important because it provides a useful, broad partitioning of the enormous class of "nonlinear systems." Thus, we discuss it briefly here and will revisit it in subsequent sections, particularly in connection with the distinction between "autoregressive" and "moving average" discrete-time models and in connection with "mixing conditions" in statistics.

The precise definition of a fading memory system given by Boyd and Chua [12] is as follows. The set $\mathbf{C}(\mathbf{R})$ is defined as the space of bounded continuous functions on the real line $\mathbf{R}$, equipped with the supremum norm $||u|| = sup_{t\in\mathbf{R}} |u(t)|$. Let K be a subset of $\mathbf{C}(\mathbf{R})$ and let N be the nonlinear operator from $\mathbf{C}(\mathbf{R})$ to $\mathbf{C}(\mathbf{R})$ defined by the nonlinear system of interest. This operator has *fading memory* on the subset K if there exists a decreasing function $w(t) : \mathbf{R}_+ \to (0, 1]$ with $lim_{t\to\infty}\, w(t) = 0$ such that for each $u \in K$ and $\epsilon > 0$, there is some $\delta > 0$ such that for all $v \in K$,

$$\begin{aligned} sup_{t\geq 0}\, |u(-t) - v(-t)|w(t) &< \delta \\ &\Rightarrow \quad |Nu(0) - Nv(0)| < \epsilon. \end{aligned}$$

Mathematically, this notion is a stronger version of continuity for the operator N, which would result if we took $w(t) = 1$ for all t [12]. Boyd and Chua prove that all finite-dimensional (exponentially stable) linear systems are fading memory systems.

The dependence on the set K in the definition of fading memory systems is important, for three reasons. First, it emphasizes the importance of explicitly considering the class of relevant inputs in modeling nonlinear systems, a point to which we shall return repeatedly. Second, and more immediately, Boyd and Chua's main approximation result is that if N has fading memory on the following specific set K:

$$K \;=\; \{u \in \mathbf{C}(\mathbf{R}) \mid ||u|| \leq M_1,\; ||u(t-\tau) - u(t)|| \leq M_2\tau,\; \tau \geq 0\;\},$$

it may be approximated with arbitrary accuracy by a finite Volterra series, for all $u \in K$. Physically, the set K consists of "slew-rate limited" signals that do not change "too rapidly" over short time intervals. In particular, piecewise-constant input sequences like steps are explicitly excluded from this set. Finally, the third point about the set K is that if N has fading memory on K and has a state-space representation, and if X is the set of states reachable from the input sequences in K, then N has a unique steady-state for all $u \in K$ and all initial conditions in X. Thus, fading memory systems are "well-behaved" in the sense that they cannot exhibit multiple steady-states or other related phenomena like chaotic responses. This point is important because chemical processes (in particular, reactors) can exhibit these phenomena. *Consequently, if fading memory models are to be developed, we can only expect them to be valid "locally," i.e., within the basin of attraction of a single steady state.*

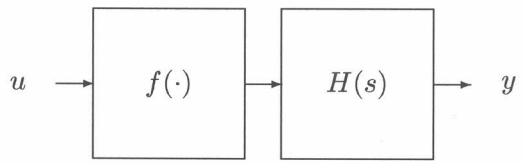

Figure 2.2: Hammerstein Model Structure

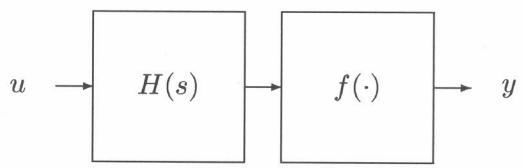

Figure 2.3: Wiener Model Structure

2.2.4 Block-Oriented Nonlinear Models

Closely related to Volterra models are the *block-oriented* models, defined by cascade and/or parallel connection of static nonlinearities and linear dynamics. Probably the best-known member of this class is the *Hammerstein model* [3, 37] shown in Fig. 2.2. Because of its relatively simple structure, this model has become increasingly popular as a "next-step-beyond-linear-modeling" of chemical processes [31, 91]. In particular, note that this model combines linear dynamics with a nonlinear steady-state gain. If the order of these blocks is reversed — i.e., if the static nonlinearity *follows* the linear dynamics — the resulting system is called a Wiener model [3, 40, 36, 38], which has also been considered for process modeling applications [99, 100].

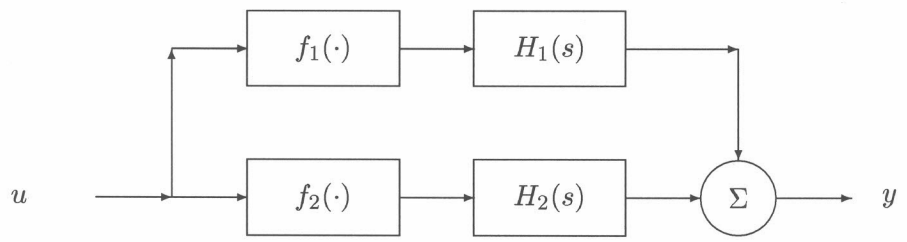

Figure 2.4: "Two channel" Uryson model

This model is shown in Fig. 2.3 and is *not* equivalent to the Hammerstein model [70], even though if $H(0) = 1$, the steady-state gain is specified by the function $f(\cdot)$ in both models and is therefore the same. Both of these structures are special cases of the more general "sandwich model" considered by Brillinger [15] and Greblicki and Pawlak [38], in which the static nonlinearity is "sandwiched" between two linear dynamic models. Still more general "block-oriented" nonlinear models have been investigated involving both series and parallel connections of static nonlinearities and linear dynamics [3, 38, 40]. As a specific example, the *Uryson model* consists of several Hammerstein models connected in parallel, driven by a common input, with their outputs summed [3], as shown in Fig. 2.4.

Chen [17] gives a reasonably detailed review of some of the recent electrical engineering and biological modeling literature on block-oriented nonlinear models. For SISO models, he introduces the following useful terminology — the letter "N" is used to indicate the placement of a static nonlinearity and the letter "L" is used to indicate the placement of linear dynamics. Thus, Hammerstein models are referred to as "NL" models, Wiener models are referred to as "LN" models, and sandwich models are referred to as "LNL" models. This nomenclature is extended to "parallel models," consisting of m of each of these models connected in parallel. Thus, the Uryson model composed of m Hammerstein models in parallel is designated "PNL_m." A number of useful results are presented in Chen's review and some of them will be discussed in subsequent sections here. It should be noted that his review does not cover any of the recent chemical engineering applications of block-oriented models, nor does it mention the growing literature on non-

parametric identification approaches for these models, a topic introduced in Section 4 of this chapter and considered further in Section 7.

In any of these models, if the static nonlinearities are analytic, a Volterra representation may be developed from the Taylor series expansion of the nonlinearity and the convolution representation of the linear dynamics. To see this connection, consider the quadratic nonlinearity $f(x) = x^2$ in both the Wiener and the Hammerstein models. If $h(t)$ is the impulse response of the linear subsystem and $k_1(t;t_1)$ is defined as in Eq. (2.13), we have for the Wiener model:

$$\begin{aligned} y(t) &= \left[\int_{-\infty}^{\infty} k_1(t;t_1)u(t_1)dt_1 \right]^2 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k_2(t;t_1,t_2)u(t_1)u(t_2)dt_1dt_2, \end{aligned} \tag{2.14}$$

where $k_2(t;t_1,t_2)$ is the separable kernel:

$$k_2(t;t_1,t_2) = k_1(t;t_1)k_1(t;t_2). \tag{2.15}$$

Similarly, it is not difficult to show that the Hammerstein model is completely described by the second-order Volterra kernel:

$$k_2(t;t_1,t_2) = k_1(t;t_1)\delta(t_1 - t_2), \tag{2.16}$$

where $\delta(\cdot)$ is the Dirac delta function.

Analogous reasoning leads to finite Volterra representations for block-oriented models with polynomial nonlinearities of arbitrary order, and from there to infinite Volterra representations for analytic nonlinearities. Note that a general characteristic of analytic Hammerstein models is that they involve "diagonal kernels" of the form:

$$k_n(t;t_1,t_2,...,t_n) = \gamma(t;t_1)\delta(t_1 - t_2)\delta(t_3 - t_2)\cdots\delta(t_n - t_{n-1}). \tag{2.17}$$

Similarly, analytic Wiener models will involve separable kernels of the general form:

$$k_n(t;t_1,t_2,...,t_n) = \alpha_n\gamma(t;t_1)\gamma(t;t_2)\cdots\gamma(t;t_n). \tag{2.18}$$

One of the topics that Chen's review does treat fairly extensively is the relationship between the different Volterra kernels that must be satisfied by the more complex block-oriented models [17].

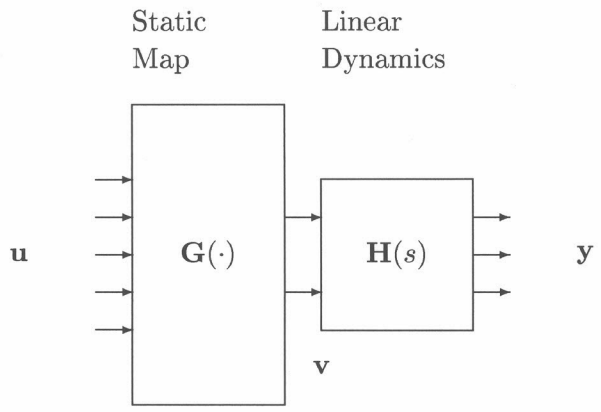

Figure 2.5: General "MIMO Hammerstein model" structure

2.2.5 MIMO Models and "Hidden Layers"

The preceeding discussions have focused on single-input/single-output models. These models have combined different forms of nonlinearity and dynamic complexity and thus occupy one face of the "process characterization cube" introduced in Section 1. To explore more of this cube, it is necessary to consider multivariable models, which are quite important in practice in spite of their added complexity. In particular, note that first-principles models are almost universally multivariable — specific control problems may focus on "single loop pairings," but this focus is an approximate one and sometimes an inappropriate one. In the linear case, multivariable state space models are simple extensions of SISO models, building on the same basic concepts of linear algebra, but these models only serve to map out the second face of the process characterization cube. To see the character of the interior of the cube, it is necessary to consider the simultaneous interplay between dynamics, nonlinearity, and multivariable character. In general, this region represents unexplored territory, but the following discussion of block-oriented MIMO models illustrates some important ideas. In addition, it will be seen that adopting a MIMO focus illuminates some important relationships between different classes of SISO models.

Block-Oriented MIMO Models

First, consider the "multivariable Hammerstein model," shown in Fig. 2.5. There, the scalar input variable u defining the standard Hammerstein model has been replaced with an m-vector $\mathbf{u}$ and the scalar output variable y has been replaced with an n-vector $\mathbf{y}$. In addition, an intermediate q-vector $\mathbf{v}$ is also shown, corresponding to the output from the first block of the diagram and the input to the second. This second block is a general q-input/n-output *linear* system, defined by the $n \times q$ transfer function matrix $\mathbf{H}(s)$. The first block is a $q \times m$ *nonlinear static map* $\mathbf{G}(\cdot)$, transforming the input vector $\mathbf{u}(t)$ at time t into an "intermediate" output vector $\mathbf{v}(t)$ at the same time instant. Each component of this "intermediate" output vector may depend arbitrarily on any or all of the components of the input vector, but there is no dependence on past inputs.

Multivariable Wiener models can be defined similarly, simply reversing the order of the two blocks appearing in Fig. 2.5; and "sandwich" or "LNL" models may be constructed by cascading two MIMO linear dynamic models with a multivariable static nonlinearity, just as in the SISO case. More complex block-oriented MIMO structures are discussed in some detail in Chen's review [17]. In all cases, the dimensions of the "internal" or "hidden" layers of the model (e.g., $\mathbf{v}$ in the MIMO Hammerstein model discussed above) represent useful "design variables," as the following subsection illustrates.

SISO Models with a "Hidden Layer"

Typical feedforward neural network structures involve an "input layer" of processing elements, one or more "hidden layers," and an "output layer" [85]. If we consider the special case of the MIMO Hammerstein model shown in Fig. 2.5 with $m = n = 1$ but $q > 1$, we see an interesting parallel: there is a q-dimensional "hidden layer" between the static nonlinear "input layer" and the linear dynamic "output layer." In particular, note that each component v_i of this "hidden layer" is defined by:

$$v_i(t) \;=\; G_i(u(t)), \tag{2.19}$$

where $G_i(\cdot) : R^1 \rightarrow R^1$ is a static nonlinearity. The "output layer" in this model is a linear MISO transfer function matrix $\mathbf{H}(s)$; that is, each component of the vector $\mathbf{v}$ passes through a linear SISO dynamic system and the results are summed to generate the response y. Combining these two layers results in the Uryson model discussed above, composed of q SISO

Hammerstein models connected in parallel, or in Chen's terminology, the PNL_q model. *An important point discussed further in Section 5.3 is that this model structure is closely related to the concept of "local linear modeling."*

If we reverse the order of the linear and nonlinear blocks in this Hammerstein model structure, we obtain a Wiener model that is also quite interesting. Specifically, for the "input layer," consider a linear state-space model of dimension $n \geq q \geq 1$ with a scalar input u and a q-dimensional output $\mathbf{v}$. This "hidden layer response" will be:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ \mathbf{v}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{2.20}$$

where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{b}$ is an n-vector, and $\mathbf{C}$ is a $q \times n$ matrix. The "output layer" then consists of the sum of q static nonlinearities, each applied to one component of $\mathbf{v}$, i.e.:

$$y(t) = \sum_{i=1}^{q} G_i(v_i(t)). \tag{2.21}$$

The two extremes of this representation are particularly interesting: taking $q = 1$, we recover the standard Wiener model, while Boyd and Chua have shown that any fading memory SISO system can be approximated arbitrarily closely by a model of this general structure with $q = n$ [12]. The question of when alternative structures might be advantageous — e.g., intermediate values of q in this "hidden layer Wiener model," the "hidden layer Hammerstein" structure considered above, "hidden layer sandwich models," or others — raise questions reminiscent of the debate over the optimal number of hidden layers and nodes in neural network approximations of static nonlinear maps [52].

2.3 Discrete-Time Process Models

While fundamental physical models are almost always developed in continuous-time, computer-based process control systems function in discrete-time: measurements are made and control actions are taken at discrete time instants, seconds, minutes, hours, or days apart. In addition, the input/output data we have available for model identification is generally only available at discrete time instants. Thus, while we can identify continuous-time models from this data, it is usually easier to identify discrete-time models and use these as

a basis for designing discrete-time control systems for computer implementation. *This observation motivates our interest in discrete-time models, despite certain inherent differences between the behavior of discrete-time models and continuous-time models.*

2.3.1 ARMAX and NARMAX Models

The class of ARMAX models (AutoRegressive Moving Average models with eXogenous inputs) has been widely used as a basis for linear model identification [55]. These models relate an input sequence $\{u(k)\}$ to an output sequence $\{y(k)\}$ by the linear, constant-coefficient difference equation:

$$\begin{aligned} y(k) &= \sum_{j=1}^{p} a_j y(k-j) + \sum_{j=0}^{q} b_j u(k-j) \\ &\quad + \sum_{j=0}^{r} c_j e(k-j). \end{aligned} \tag{2.22}$$

This model originated in the time-series literature where the focus is on the statistical characterization of the output sequence $\{y(k)\}$ resulting when $\{e(k)\}$ is a "Gaussian white noise" sequence (see Section 6). The term "exogeneous input" for the sequence $\{u(k)\}$ refers to the fact that the ARMAX model is a generalization of the ARMA model obtained when $u(k) = 0$ identically. In the process control applications considered here, the emphasis is somewhat different. Specifically, it is the relationship between the sequences $\{u(k)\}$ and $\{y(k)\}$ that is of primary importance, while the sequence $\{e(k)\}$ represents a "modeling error" in this relationship, arising from the combined effects of measurement noise, unmeasured process disturbances, neglected nonlinearities, etc. The assumptions made regarding the nature of the two "model inputs" $\{u(k)\}$ and $\{e(k)\}$ will influence different aspects of the modeling problem in different ways and will be discussed from various perspectives in subsequent sections.

For nonlinear systems, a popular class of discrete-time models is the NARMAX (Nonlinear ARMAX) family, described by Billings and Voon [5]:

$$\begin{aligned} y(k) &= F(\, y(k-1), y(k-2), ..., y(k-p), \\ &\qquad u(k), u(k-1), ..., u(k-q) \\ &\qquad e(k-1), e(k-2), ..., e(k-r)) \; + e(k). \end{aligned} \tag{2.23}$$

Here, $F(\cdot)$ is a nonlinear function of the $p + q + r + 1$ variables indicated and the sequences $\{y(k)\}$, $\{u(k)\}$, and $\{e(k)\}$ have the same definitions as in the ARMAX model described above. These models constitute an extremely broad class, including many other classes of nonlinear discrete-time models as special cases. To define a NARMAX model completely, it is necessary to specify the *order parameters* p, q, and r, and the form of the function $F(\cdot)$ appearing in Eq. (2.23).

In this section, we will focus on ideal input/output behavior for which $e(k) = 0$ identically, implying $r = 0$ by default. In contrast, the order parameters p and q are quite important. In particular, taking $p = 0$ and $q > 0$, the model output $y(k)$ depends only on the past history of the input sequence $\{u(k)\}$ and not on previous output values $y(k - j)$. Borrowing terminology from the time-series literature [76, 96], we will call this class the "Nonlinear Moving Average models with eXogeneous inputs," or NMAX models. Conversely, if we take $p > 0$ and $q = 0$, we obtain a model whose output $y(k)$ depends only on the current input $u(k)$ and the past history of the *output* sequence $\{y(k - j)\}$. Again borrowing terminology from the time-series literature, this class will be called the "Nonlinear AutoRegressive models with eXogeneous inputs" or NARX models. As a practical matter, $y(k)$ will generally depend on the input $u(k - d)$, delayed by a *single, fixed* number of samples d; this assumption does not change the nature of the model class — they may still be viewed as NARX models. What is *not* permitted in the NARX model class considered here is dependence on *several different* past inputs. If both $p > 0$ *and* $q > 0$, the output $y(k)$ at any time k will depend on both the input history and the output history, giving rise to the general NARMAX class, including both the NMAX and the NARX classes as special cases and exhibiting a correspondingly richer variety of dynamics. (It should be noted that Ljung [55] defines linear ARX models in a way that permits inclusion of "moving average" terms in the control input $u(k)$; due to the substantial qualitative behavior differences between nonlinear "autoregressive" and "moving average" models discussed below, we will restrict NARX models to exclude terms that are "moving average" with respect to the control input, as noted above.)

2.3.2 Non-Equivalent Representations

One unique feature of the linear problem is that if a continuous-time linear ordinary differential equation with constant coefficients is sampled at a constant sampling rate T_s, the samples may be described exactly by a discrete-

time linear model [55, p. 20]. In addition, if T_s is sufficiently small to avoid "aliasing phenomena," this transformation is invertible. That is, given T_s and the discrete-time ARMAX model (2.22), we may derive an N^{th}-order constant-coefficient, linear ordinary differential equation relating $u(t)$ and $y(t)$ such that $y(t_k) = y(k)$ and $u(t_k) = u(k)$ for $t_k = (k-1)T_s$. Another characteristic feature of linear discrete-time ARMAX models is that they may be represented equally well by either "moving average" or "autoregressive" expressions. Specifically, note that the case $p = 0, q = \infty$ corresponds to a convolution model with $\{b_j\}$ representing the impulse response of the system. It is a standard result [55, p. 13] that any time-invariant, linear system may be completely characterized by its impulse response. Conversely, the case $p = \infty$, $q = 0$ represents an infinite-order autoregressive model, and it is not difficult to show that any time-invariant, linear system may be completely characterized by a model of this form as well [49, p. 112]. In other words, the response of any stable linear system may be represented either as a convolution model in terms of the past history of its *input* sequence or as an autoregressive model in terms of the past history of its *output* sequence.

These equivalences for linear models — continuous, discrete, autoregressive, and moving average — are summarized in Fig. 2.6. Unfortunately, *none* of these equivalences extend to nonlinear problems. For example, the logistic equation:

$$y(k) = ay(k-1)[1 - y(k-1)], \tag{2.24}$$

is a first-order nonlinear discrete-time model that has been extensively studied because it exhibits chaotic behavior [96]. In contrast, it is known that in continuous time, nonlinear ordinary differential equations must be at least third-order to exhibit chaotic behavior [39].

To see that NMAX and NARX models are not equivalent, consider the following NARX model [96]:

$$y(k) = \begin{cases} 2y(k-1) + u(k) & |y(k-1)| \leq 2 \\ u(k) & |y(k-1)| > 2 \end{cases} \tag{2.25}$$

If this model is driven by the input sequence $u(k) = (-1)^k$ — a periodic sequence with period 2, the response is a sequence with period 6. This lengthening of the period of the response relative to the period of the input is called *subharmonic generation* and is not possible in NMAX models [70]. That is, note that the response of an NMAX model to an input sequence $\{u(k)\}$ with period T is of the form:

$$y(k) = F(u(k), u(k-1), ..., u(k-m))$$

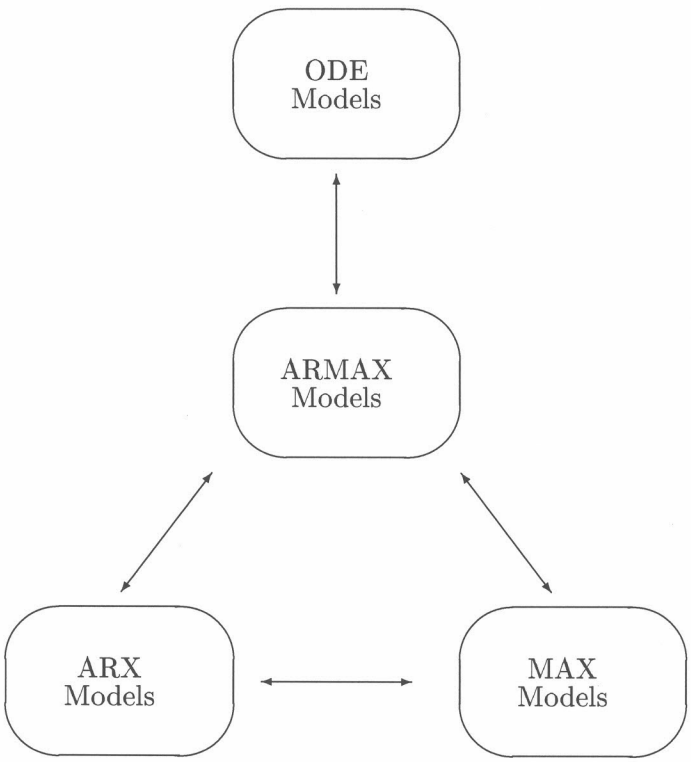

Figure 2.6: Equivalent Linear Models

$$= F(u(k+T), u(k+T-1), ..., u(k+T-m))$$
$$= y(k+T). \tag{2.26}$$

Thus, regardless of the form of the nonlinear function $F(\cdots)$, the response sequence $\{y(k)\}$ has the same period as the input sequence $\{u(k)\}$, excluding the possibility of subharmonic generation.

Since all of the equivalences noted in Fig. 2.6 hold for linear models, linear model structure selection is therefore largely a matter of convenience — it may be of substantial practical importance, but selecting the "wrong" model structure does not exclude any *qualitative* phenomena. Conversely, since *none* of these model structure equivalences hold for the nonlinear case, we must be more judicious in nonlinear model structure selection.

2.3.3 Special Classes of NARMAX Models

As noted above, the class of NARMAX models is an extremely large one, and in practice, it is usual to focus on various structurally defined subsets of this class. The first two steps in the four-step model-building procedure described in Section 1 then entail selection of a specific sub-class of NARMAX models and the specification of order parameters p, q, and r in Eq. (2.23). The selection of a sub-class of NARMAX models amounts to specifying a general form for the function $F(\cdots)$ in Eq. (2.23), a topic to be discussed further in Section 4. To facilitate that discussion, it is useful to consider first, briefly, the following special classes of NARMAX models.

NAARX Models

In theory, one approach to the model structure selection problem is the use of nonparametric statistical procedures, discussed briefly in Section 4. In practice, however, these procedures become increasingly difficult to apply as the number of variables involved increases. In Eq. (2.23), this number is $p + q + r + 1$, which is large enough to be a significant concern even for low-order models. Consequently, we must either impose a parametric structure on $F(\cdot)$ *a priori* or decompose the general structure into a collection of simpler sub-structures to which nonparametric techniques are applicable. The first approach is probably the most common in practice, typically taking $F(\cdot)$ as a polynomial of relatively low degree ℓ. The advantage of this choice is that it permits the use of statistical techniques like stepwise regression for

model identification [3]. More recently, Billings and Zhu [7, 102] have applied similar procedures to the identification of *rational NARMAX* models, in which $F(\cdot)$ is the ratio of two polynomials. Alternatively, Tong [96] considers a variety of different models with the NARMAX structure in which the function $F(\cdot)$ is not a polynomial but instead involves discontinuous threshold functions.

The second alternative is illustrated by the class of NAARX models (Nonlinear Additive AutoRegressive models with eXogeneous inputs) defined by Chen and Tsay [16]:

$$y(k) \;=\; \sum_{i=1}^{p} f_i(y(k-i)) \;+\; \sum_{j=0}^{q} g_j(u(k-j)) \;+\; e(k). \qquad (2.27)$$

Here, the functions $\{f_i(\cdot)\}$ and $\{g_j(\cdot)\}$ are scalar nonlinearities, making them well suited to the use of nonparametric procedures like those discussed in Section 4. This class of models is broad enough to exhibit a very wide range of behavior. In particular, note that this class includes the linear ARMAX models, along with many interesting nonlinear model classes considered later in this chapter. For example, note that both the logistic equation (2.24) and Tong's subharmonic generator model (2.25) are members of this class. Conversely, this model does not permit "cross-terms" involving products of inputs and outputs, or input or output values at different times. *An extremely interesting open question is what behavioral price we pay for imposing this analytically convenient structural restriction on a NARMAX model.*

Volterra and Block-Oriented Models

Discrete-time Volterra models may be defined by replacing the "multiple convolution integrals" appearing in Eq. (2.12) with "multiple convolution sums," i.e.:

$$\begin{aligned} y(k) = y_0 \;+\; & \sum_{j=0}^{\infty} a_j u(k-j) \qquad (2.28) \\ & + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} b_{i,j} u(k-i)u(k-j) \\ & + \sum_{l=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} c_{l,i,j} u(k-l)u(k-i)u(k-j) \;+\; .... \end{aligned}$$

As in the case of continuous-time Volterra models, practical applications require that this series be truncated to a finite number of terms; these truncated series are known as *finite Volterra models.*

Extending their continuous-time results, Boyd and Chua [12] also define discrete-time fading memory systems and prove that these systems may be approximated arbitrarily well by finite Volterra models. Specifically, they consider nonlinear system operators N mapping ℓ^∞ into itself, where ℓ^∞ is the space of bounded sequences $\{u(k)\}$ equipped with the supremum norm $||u|| = sup_k\ |u(k)|$. The operator N has fading memory on a subset K of ℓ^∞ if the following conditions are met. First, there is a decreasing sequence $\{w(k)\}$ such that $0 < w(k) \leq 1$ for all k and $lim_{k\to\infty}\ w(k) = 0$. The operator N then has fading memory on K if, for each $u \in K$ and each $\epsilon > 0$, there is some $\delta > 0$ such that for all $v \in K$,

$$\begin{aligned} sup_{k\geq 0}\ |u(-k) - v(-k)|w(k) &< \delta \\ &\Rightarrow\ |Nu(0) - Nv(0)| < \epsilon. \end{aligned} \tag{2.29}$$

As in the continuous-time case, this condition implies that "the distant past has negligible influence on the present." Two important points are worth noting here, however. First, unlike the continuous-time case, the finite Volterra approximation result holds here for any set K of uniformly bounded sequences: i.e., $||u|| \leq M_1$. In particular, note that no "slew-rate limit" is imposed here, suggesting the discrete-time result may have wider validity than the continuous-time result. Conversely, the second point is that, as Boyd and Chua note [12], these finite Volterra models belong to the moving average class — i.e., they are polynomial NARMAX models with $p = 0$. Also, as in the case of continuous-time fading memory models, discrete-time fading memory models cannot exhibit output multiplicities [12]. It is known that the CSTR model can exhibit such multiple steady-state behavior [97], so it follows that any fading memory model (e.g., linear, Volterra, etc.) can only be a "reasonable" approximation over an operating range that is narrow enough to include just one steady-state. *This observation raises the interesting open question: what class of physical phenomena correspond to discrete-time models that do* **not** *have fading memory on the set $\mathcal{S}$ of input sequences of primary interest?*

Even when q is relatively small and only a few terms are retained in a finite Volterra model, the number of model parameters required to represent the series quickly becomes unreasonable. To address this problem in practice, various approaches to structural restriction have been proposed,

including "pruning" (i.e., setting certain Volterra coefficients to zero) [74] and the use of orthogonal polynomial expansions [57]. Alternatively, this objection may be overcome by considering instead the discrete-time versions of the block-oriented models discussed in Section 2. As in the continuous-time case discussed there, models constructed from analytic nonlinearities may be viewed as structurally constrained Volterra models. Conversely, the block-oriented structure remains well-defined even if these nonlinearities are not analytic. This observation may be important in practice since the nonlinearity $f(\cdot)$ in either the Hammerstein or the Wiener model represents the steady-state gain of the model and analyticity is a fairly restrictive condition. For example, note that analytic functions cannot exhibit *hard saturation*: if $f(x) = c$ on some finite interval $[a, b]$ but this function is not identically constant, it is not analytic. This result follows from the fact that the behavior of an analytic function on any open interval completely determines its behavior [19, p. 87]. To distinguish between Hammerstein models with analytic nonlinearities from those with more general nonlinearities, it is useful to define the sub-class of *analytic-* or *A-Hammerstein* models. Similarly, Wiener models with analytic nonlinearities will be referred to as *analytic-* or *A-Wiener* models.

"Cross-Terms" and the PPOD Model

It is useful to note that the discrete-time Hammerstein model is a member of the NAARX class defined in Eq. (2.27). To see this connection, take $g_0(x) = f(x)$ as the static nonlinearity in the Hammerstein model and represent the linear dynamic part of the model by an ARMAX model with coefficients $\{a_i\}$ and $\{b_i\}$. The other functions appearing in the NAARX representation are then $f_i(x) = a_i x$ and $g_i(x) = b_i g_0(x)$. Conversely, note that Wiener models are *not* members of the NAARX class: "cross-terms" are present in the Wiener model. Again, an extremely interesting issue, both theoretically and practically, is what we gain or lose by including or excluding these terms from the general NARMAX structure.

Partially motivated by this question, Pawlak, Doyle and ourselves [69] have proposed a generalization of the Hammerstein model in which the nonlinearity itself is dynamic. This model structure will be abbreviated as the "PPOD model" and is of the form shown in Fig. 2.7. It is described by the

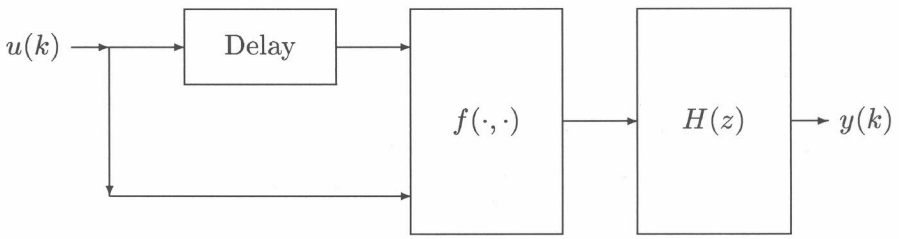

Figure 2.7: "PPOD" Model Structure Proposed by Pawlak, Pearson, Ogunnaike, and Doyle

following equation:

$$y(k) = \sum_{i=1}^{p} a_j y(k-j) + \sum_{j=0}^{q} b_j m(u(k-j), u(k-j-1)), \quad (2.30)$$

where $m(\cdot,\cdot) : R^2 \rightarrow R$ represents a dynamic nonlinearity, involving the lowest-order "cross-terms" missing from the NAARX models. In addition, by restricting the domain of this nonlinearity to R^2, we are still able to apply nonparametric regression procedures to gain some "hints" as to the nature of the cross-terms required to explain the input/output data [69].

It is instructive to consider some of the interesting special cases that are included in this general model class. First, note that if $m(x,y) = f(x)$ or $m(x,y) = g(y)$, this model reduces to the Hammerstein model. On the other hand, note that while the Hammerstein model is a member of the NAARX class, the PPOD model is not, unless $m(x,y) = f(x) + g(y)$. As a specific example, note that if $m(x,y) = g(ax + by)$, the result is a Wiener model consisting of a two-term moving average linear model followed by the static nonlinearity $g(\cdot)$. Finally, Zhu and Seborg [102] recently introduced a model of this form with $m(x,y) = \alpha x + p(y)$ where $p(y)$ is a polynomial. This model was motivated by practical considerations in implementing nonlinear model predictive control algorithms based on Hammerstein models and is discussed further in Section 4.4.

Relationships Between Different Model Classes

To conclude this section, Figs. 2.8 and 2.9 are Venn diagrams that attempt to illustrate the relationships between the different discrete-time nonlinear dynamic model classes considered here. These diagrams are based on similar ones appearing in [71], which discusses these relationships in somewhat greater detail. Note that the abbreviation "NAMAX" appearing in Fig. 2.9 refers to "Nonlinear Additive Moving Average models with eXogeneous inputs," defined by taking $p = 0$ in Eq. (2.27).

2.3.4 Added Complexity of MIMO Models

Mathematically, it is easy to extend the NARMAX model defined in Eq. (2.23) to the multiple-input/multiple-output (MIMO) setting. To handle multiple inputs, the arguments $u(k), u(k-1), ..., u(k-q)$ in the function $F(\cdot)$ are replaced with the larger set $u_1(k), u_1(k-1), ..., u_1(k-q_1), u_2(k), u_2(k-1), ..., u_2(k-q_2), ...$, etc. To handle multiple outputs, one equation of the form (2.23) is written for each output $y_i(k)$, with the arguments of each function $F_i(\cdot)$ further augmented to include the terms $y_j(k-1), y_j(k-2), ..., y_j(k-p_j)$ for the "other outputs" $j \neq i$. This generalization leads to combinatorial growth in model complexity since the number of *arguments* in each function $F_i(\cdot)$ for an n-input, m-output model will be $n_{arg} = p_1 + p_2 + \cdots + p_m + (q_1 + 1) + (q_2 + 1) + \cdots + (q_n + 1)$, assuming the past model error terms $e(k), e(k-1), ..., e(k-r)$ are omitted (as they often are in practice).

Even if each function $F_i(\cdot)$ is restricted to be a low-order polynomial, the number of possible terms in this model quickly becomes prohibitive. For example, consider a three-input, two-output model with $p_i = 2$ for all i and $q_j = 2$ for all j — each of the two functions $F_i(\cdot)$ then involves 13 arguments. Even if these functions are restricted to be cubic polynomials, the number of *possible* terms in each polynomial is 455, meaning that in the general case 910 model coefficients must be estimated to specify both functions fully. Increasing the polynomial order to 4 increases the total number of model coefficients to 14,560. In contrast, in the SISO case, for $p = 2$ and $q = 2$, $F(\cdot)$ is a function of 5 arguments and the total number of unknown model parameters is 35 for the cubic polynomial and 70 for the fourth-order polynomial case. Thus, as important as judicious model structure selection is in the SISO case, it is even more critical in the MIMO case. This particular point is illustrated in the case study in Section 9, in which a MIMO NARMAX model is developed for a high-purity distillation

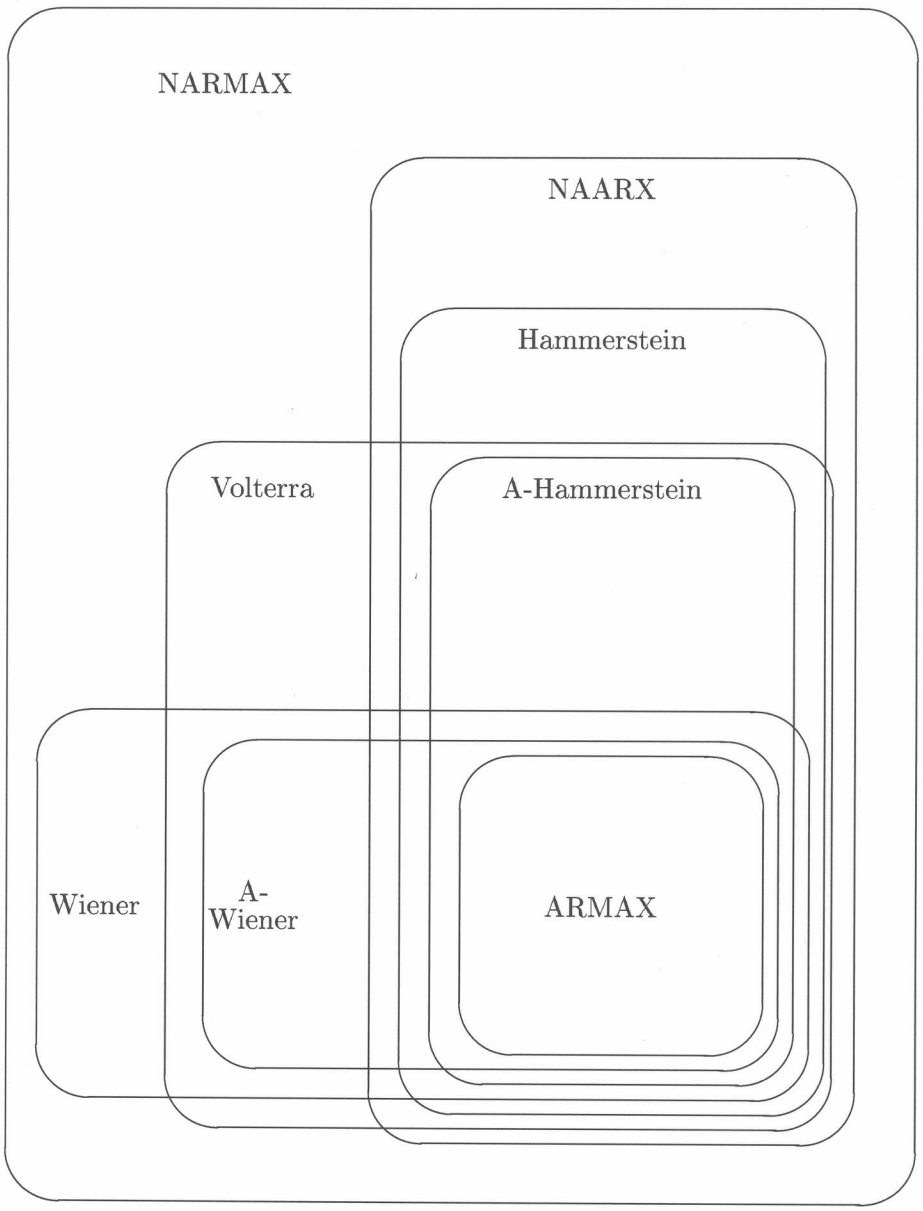

Figure 2.8: General taxonomy of NARMAX models

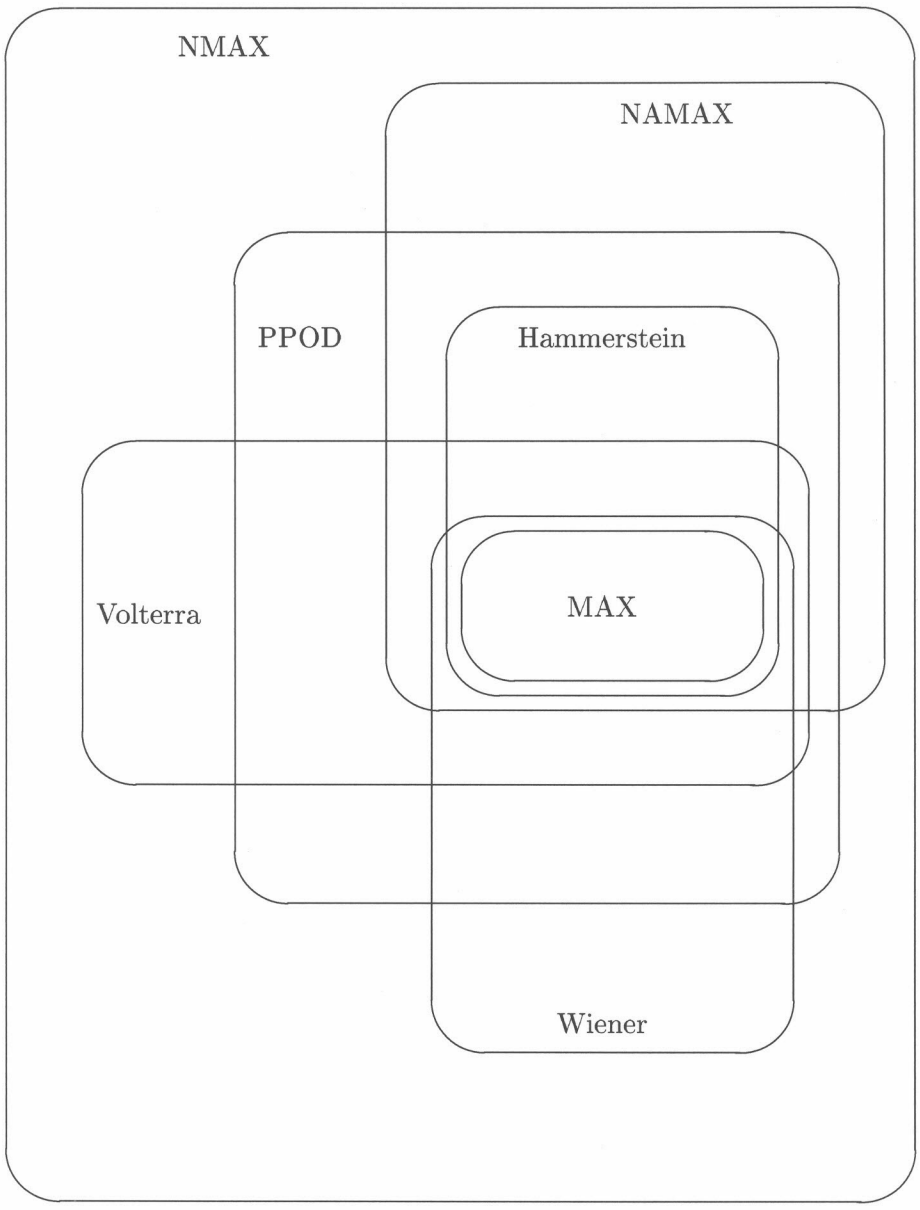

Figure 2.9: Taxonomy of NMAX models

column. One possible approach to the multivariable NARMAX modeling problem is to note its similarity to the multivariable regression problem [28]. There, polynomial models in which all possible terms are present are called *saturated models* [78] and approaches like stepwise regression [5, 44] have been developed to arrive at non-saturated models. "Regression-based" approaches like these will be considered briefly in Section 7 in connection with NARMAX model identification.

2.4 The Empirical Model-Building Process

The fundamental objective of empirical modeling is to construct a mathematical model $\mathcal{M}$ of a physical process $\mathcal{P}$. Given a set $\mathcal{S}$ of input sequences, a necessary — *but not sufficient* — condition for $\mathcal{M}$ to be a "good" model is that the predicted response $\mathcal{M}\mathbf{u}$ for any input $\mathbf{u} \in \mathcal{S}$ is approximately equal to the response $\mathcal{P}\mathbf{u}$ of the physical process to the same input. In symbols, we are seeking "solutions" $\mathcal{M}$ to the approximate equation:

$$\mathcal{M}\mathbf{u} \simeq \mathcal{P}\mathbf{u} \text{ for all } \mathbf{u} \in \mathcal{S}. \tag{2.31}$$

The "solution" we obtain for this "equation" will depend on the following factors:

- the process $\mathcal{P}$;
- the exact interpretation of the approximation symbol "$\simeq$";
- the set $\mathcal{S}$ of process inputs;
- the class $\mathcal{C}$ of empirical models $\mathcal{M}$ from which we are seeking the "best" representative.

Here, we assume that the process $\mathcal{P}$ is fixed, but that the other three items on this list are at least partially at the discretion of the model builder. Section 4.1 illustrates some of our options in interpreting the symbol "$\simeq$", while Section 4.2 considers specification of the set $\mathcal{S}$ of inputs over which we want our model to be valid. In practice, model-building is an iterative process in which we attempt either to "tighten" our notion of approximation, or to "enlarge" the model validity set $\mathcal{S}$. Section 4.3 briefly discusses this iteration process. The class $\mathcal{C}$ of candidate models to be considered must be specified at each stage of this iteration process; Section 5 is devoted to a detailed discussion of this model structure selection problem.

2.4.1 Interpretation of "$\simeq$"

The range of possible interpretations for the approximation symbol "$\simeq$" — and the impact of our choice on the final modeling result — is most easily illustrated with a simple example. Specifically, consider the problem of approximating an analytic function $f(\cdot)$ on a specified interval $[a, b]$. Three different "reasonable" approximation strategies will be considered and shown to lead to fundamentally different results: Taylor series expansion, Tchebyshev polynomial approximation, and Bernstein polynomial approximation.

Taylor Series Approximations

Since $f(\cdot)$ is analytic, it may be expanded as a Taylor series, i.e.:

$$f(x) = f(x_0) + (x - x_0)\frac{df}{dx}\bigg|_{x=x_0} + (x - x_0)^2\frac{d^2f}{dx^2}\bigg|_{x=x_0} + \ldots, \quad (2.32)$$

where x_0 is some reference point in the interval $[a, b]$. If we are to use this expansion in practice, it must be truncated to a finite number of terms n, resulting in an approximation error that depends on n. It is important to note that the underlying approximation philosophy here is a *local* one; in particular, the expansion is *exact* at $x = x_0$ and degrades as we move away from this point. To increase the approximation accuracy of the Taylor series on the interval $[a, b]$, we must increase the number of terms retained. Some Taylor series converge very slowly, however, such as the one for $f(x) = (1 + x)^{-1}$, valid for the range $-1 < x < 1$, but *very* slowly convergent as the limits of this range are approached.

Tchebyshev Polynomial Approximations

In contrast to the local nature of Taylor series expansions, Tchebyshev polynomials solve the problem of minimizing, for a fixed interval $[a, b]$ and a fixed polynomial order n, the maximum approximation error incurred on the interval [25, sec. 7.4]. That is, the Tchebyshev polynomial approximation $T_n(f; x)$ minimizes the maximum value of $|f(x) - T_n(f; x)|$ on the interval $[a, b]$. Thus, for given order n, the worst-case approximation error for the Tchebyshev polynomial approximation $T_n(f; x)$ is better than the worst-case Taylor series approximation error. Conversely, the *derivatives* of the function $f'(x)$, $f''(x)$, etc. are generally *not* well approximated by the corresponding derivatives of $T_n(f; x)$. This behavior is in contrast to the *local* behavior of

the Taylor series, since the lowest-order derivatives of the truncated Taylor series are also exact at x_0, just like the function itself.

Bernstein Polynomial Approximations

By the Weierstrass approximation theorem [50], any continuous function $f(x)$ can be approximated uniformly to arbitrary accuracy on any finite, closed interval $[a, b]$ by a polynomial of finite degree. The polynomials often used to prove this theorem are the Bernstein polynomials $B_n(f; x)$, given by the explicit formula [25]:

$$B_n(f;x) = \sum_{k=0}^{n} f(k/n) \begin{pmatrix} n \\ k \end{pmatrix} x^k (1-x)^{n-k}. \tag{2.33}$$

(This formula assumes the interval $[a, b]$ is $[0, 1]$; nothing significant is changed by re-scaling this expression to arbitrary finite intervals.)

It is important to note that the basis of the Weierstrass theorem is fundamentally different from the basis for Tchebyshev approximation. Specifically, Tchebyshev approximation constructs a *best approximation of fixed order n*, given the function and the approximation interval. On the other hand, in proving the Weierstrass approximation theorem, the maximum approximation error over the interval is specified and a Bernstein polynomial of *sufficiently high order n* is constructed to meet this approximation requirement. In fact, an unfortunate feature of the Bernstein polynomial approximations is that their convergence is rather slow [25], so the required polynomial order tends to be high. The primary advantage of these polynomials, however, is that the approximations preserve the *qualitative* character of the functions they approximate. In particular, low-order derivatives of Bernstein polynomial approximations are reasonable approximations of the derivatives of $f(x)$ and the approximation also preserves such general qualitative behavior as convexity and monotonicity [25].

Deciding Between Different Approximation Criteria

The key point of this discussion is that the "best nonlinear approximation" we obtain for the function $f(x)$ depends on both our "adequacy" criterion and the range over which we want this criterion to be met. In the above example, if our sole criteria is "goodness of fit," we will prefer the Tchebyshev approximation over either the Taylor series or the Bernstein polynomial approximation because, for a specified interval and polynomial order, it has

the smallest possible worst-case approximation error. Conversely, the truncated Taylor series has a certain "fundamental feel" about it, since each term in the expansion is directly related to the derivatives of the function at a specified point. Note, however, that a truncated Taylor series expansion need not exhibit the general qualitative behavior of the function $f(x)$ on a given interval $[a, b]$. For example, consider the Taylor series expansion for the function $f(x) = (1+x)^{-1}$ on the interval $[-3/4, 3/4]$; expanded about $x = 0$ and truncated to second order, this series is $1 - x + x^2$. This approximation has a minimum in the interior of the approximation interval at $x = 1/2$, even though the original function $f(x)$ is monotonically decreasing on the whole interval. If we wish to achieve an approximation that retains these qualitative features over the entire interval of interest, we must accept a more complex model like the Bernstein polynomial approximation discussed above.

2.4.2 Specification of the Range of Model Validity

Overall, the "most reasonable interpretation" of the approximation symbol "$\simeq$" in Eq. (2.31) will depend on the following factors:

- the set $\mathcal{S}$ of process inputs;
- the class $\mathcal{C}$ of empirical models under consideration;
- our application (i.e., the process $\mathcal{P}$ and the control system design methodology under consideration).

The following paragraphs illustrate this dependence on the set $\mathcal{S}$ of process inputs, demonstrating the subtlety required in adequately defining this set. Dependence on the class $\mathcal{C}$ of models is related to the issue of model structure selection considered in Section 5. Finally, note that it is application-depencence that motivates recent work on "control-relevant" model identification [81]: "open loop" prediction error measures are replaced with frequency-weighted prediction error measures that penalize prediction errors with adverse control implications more than those with negligible control implications. It is not immediately obvious how to generalize these ideas to arbitrary nonlinear model identification problems because frequency-domain descriptions of nonlinear models are necessarily incomplete. *Still, the basic concept of making our approximation measure application-dependent seems like a useful one and this area appears to be a fruitful one for future research.*

Specification of the "range of validity" for a dynamic model $\mathcal{M}$ is more complex than specification of the range of a function $f(\cdot)$ mapping one finite-dimensional space into another. Specifically, note that the set $\mathcal{S}$ appearing in Eq. (2.31) is a set of *functions* $u(t)$ rather than a set of *values*. To illustrate this distinction, recall from the discussion of fading memory systems in Section 2.3 that Boyd and Chua's approximation result [12] was valid for a set K of slew-rate limited inputs. This restriction was necessary to make the Stone-Weierstrass approximation theorem apply to fading memory systems. The key point is that this restriction specifies *two* types of constraints on u:

1. the static amplitude limit $||u(t)|| \leq M_1$;
2. the dynamic slew-rate limit $||u(t-\tau) - u(t)|| \leq M_2\tau$.

This observation is important because specification of an input range condition like (1) alone is generally not sufficient to define the set $\mathcal{S}$. The following example provides a detailed description of another case where dynamic constraints are both important and non-obvious without further examination.

A Simple Nonlinear Continuous-Time Model

Consider the behavior of the first-order SISO model obtained by approximating the function $F(x(t), u(t))$ in Eq. (2.1) by the following *second-order* polynomial [65]:

$$F(x,u) \simeq (\phi_0 + \phi_1 x + \phi_2 x^2) + (\gamma_0 + \gamma_1 x + \gamma_2 x^2)u. \tag{2.34}$$

Note that this model is control affine but not bilinear unless $\gamma_2 = 0$ and $\phi_2 = 0$. This model structure was motivated by a desire to explore the "next-step-beyond-bilinear-models," retaining the first *two* terms of the Taylor series expansion of the nonlinearities appearing in the control-affine structure.

To obtain insights into the qualitative behavior of this model, it is useful to employ the same "trick" invoked earlier for the bilinear model. Specifically, note that the step response of this model is given by the solution of an equation of the form:

$$\frac{dx}{dt} = a + bx + cx^2, \tag{2.35}$$

where:

$$a = \phi_0 + \alpha\gamma_0$$

$$\begin{aligned} b &= \phi_1 + \alpha\gamma_1 \\ c &= \phi_2 + \alpha\gamma_2. \end{aligned} \tag{2.36}$$

An explicit solution to this equation may be obtained, but its form depends strongly on the constants a, b, and c. In particular, the character of the solution depends on the discriminant $\Delta = 4ac - b^2$: if this quantity is negative, nothing remarkable happens, but if it is positive, this model can exhibit *finite escape time*. That is, for $\Delta > 0$, the solution to Eq. (2.35) is:

$$x(t) = \frac{\sqrt{\Delta}}{2c}\left[\tan\left(\theta_0 + \frac{\sqrt{\Delta}}{2}t\right) - \tan\theta_0\right], \tag{2.37}$$

where $\theta_0 = \arctan(b/\sqrt{\Delta})$. Note that this solution diverges to $+\infty$ as the argument of the tangent function approaches $\pi/2$, which occurs at the escape time:

$$t^* = \frac{2}{\sqrt{\Delta}}\left[\frac{\pi}{2} - \theta_0\right]. \tag{2.38}$$

An extremely important point to note here is that the dominant qualitative behavior of this model — presence or absence of a finite escape time — depends on both the model structural parameters ϕ_i and γ_i and the input amplitude α. To see this point, expand Δ explicitly as:

$$\Delta = [4\phi_0\phi_2 - \phi_1^2] + [4(\gamma_0\phi_2 + \gamma_2\phi_0) - 2\gamma_1\phi_1]\alpha + [4\gamma_0\gamma_2 - \gamma_1^2]\alpha^2. \tag{2.39}$$

To force $\Delta < 0$ — and thus avoid the finite escape phenomenon — the following criteria must be met. First, to satisfy this condition for small input amplitudes ($\alpha \simeq 0$), the first term in brackets must be negative, representing an "inherent stability constraint." Similarly, to avoid finite escape at large input amplitudes ($\alpha \to \infty$), the last term in brackets must also be negative.

Note that both of these conditions are satisfied by the linear special case $\phi_0 = \phi_2 = 0$ and $\gamma_1 = \gamma_2 = 0$. In fact, for this case, $\Delta = -\phi_1^2 < 0$ for all α, correctly indicating that the linear case cannot exhibit finite escape time, regardless of the input amplitude. Similarly, note that both of these conditions are also satisfied for the bilinear case, for which the constraint $\gamma_1 = 0$ is relaxed. In fact, for the bilinear case, we have $\Delta = -(\phi_1 + \gamma_1\alpha)^2$, which is, again, non-positive for all possible input amplitudes and system parameters. Thus, the bilinear special case of this model cannot exhibit finite escape time, either. Conversely, note that if the first term in brackets is negative but the last term is positive, the model will exhibit a

finite escape time for some sufficiently large input step amplitude α. As a specific example, consider $\phi_0 = 0$, $\phi_1 = 1$, $\phi_2 = 0$, $\gamma_0 = 1$, $\gamma_1 = 0$, and $\gamma_2 = 1$. Here, $\Delta = 4\alpha^2 - 1$, so the model cannot exhibit finite escape time for $\alpha < 1/2$, but it can for $\alpha > 1/2$. Finally, note that a perverse kind of "conditional stability" is also possible in this model: if the first and last terms in brackets are both negative but the middle term is positive, there may be an "intermediate amplitude" α that causes the middle term to dominate. In this case, $\Delta > 0$ and the model response will diverge in finite time. As a specific example, consider $\phi_0 = 0$, $\phi_1 = 1$, $\phi_2 = 1$, $\gamma_0 = 1$, $\gamma_1 = 1$, $\gamma_2 = 0$: it is easy to show that $\Delta > 0$ for α between $3 - \sqrt{8} \simeq 0.172$ and $3 + \sqrt{8} \simeq 5.828$.

Now, suppose we perform the following identification experiment: we wish to model a process $\mathcal{P}$ that is both well-behaved (i.e., does not exhibit finite escape times) and well approximated by a control-affine fundamental model. Suppose we excite this process with inputs from a set $\mathcal{S}$ and attempt to estimate the corresponding parameters for model (2.34) that best match the available data. Here, the set $\mathcal{S}$ consists of "high-frequency" piecewise-constant input sequences that assume arbitrary values in the interval $[-1, 1]$, but whose values switch rapidly enough that they never remain constant longer than 0.2 time units. For these inputs, the "conditionally stable" model just described is perfectly well-behaved — the worst-case (i.e., shortest) escape time for step inputs in the interval $[-1, 1]$ is $t^* = \pi/4 \simeq 0.7854$. Thus, it is possible that this model could be the "best fit" approximation of $\mathcal{P}$ from the available data, even though this model exhibits finite escape time but the process itself does not. *In particular, the point of this example is that it is the nature of the set $\mathcal{S}$ we choose that determines whether the model considered here is a "good" or a "pathologically poor" approximation of the process.* Specifically, the model considered here may be a good approximation on the set $\mathcal{S}$ of "high-frequency" piecewise-constant sequences used for identification, but it is completely inadequate when considered in terms of step responses since any positive step of amplitude between $3 - \sqrt{8}$ and $3 + \sqrt{8}$ will exhibit the finite escape instability. Note that the difference between the set $\mathcal{S}$ and the larger set of "all piecewise- constant input sequences" is that $\mathcal{S}$ excludes "low-frequency" signals (e.g., steps).

2.4.3 Sequential Model Building

In practice, model building is usually a sequential process, in which models of increasing complexity are developed from — and compared with — ear-

lier models of lower complexity. In the development of nonlinear, dynamic models, two particularly useful and important points of comparison are:

- static (i.e., steady-state) models;
- linear dynamic models.

The nonlinear model structure selection problem will be considered in Section 5, but before considering this topic, it is worth considering the two "precursors" listed above, *specifically in their role as "guides" for the iterative nonlinear modeling process.*

Steady-State Behavior of Empirical Models

The idea of "steady-state agreement" between a candidate model and process behavior is often an important criterion for "model validity," but this issue is more subtle for nonlinear models than for linear ones. In particular, for linear models, the issue is whether the steady-state gain K is correct, but for nonlinear models, the notion of "steady-state agreement" between a model and a process is closely related to the issues of local *vs.* global modeling considered in Section 5. That is, for a linear SISO problem, the process input is typically represented as $U + u(t)$ where U is a steady-state reference value and $u(t)$ is the instantaneous deviation of the input from this reference value. Similarly, the process response is represented as $Y + y(t)$, where Y is steady-state response to the input U and $y(t)$ is the corresponding instantaneous deviation from this reference value. For stable, linear, time-invariant models, the following conditions hold:

$L1$ the steady-state response is $Y = KU$, independent of $u(t)$;

$L2$ the dynamic response is $y(t) = L\{u(t)\}$, independent of U,

where L is the linear model's dynamic response.

For nonlinear models, the situation is more complex, and we have the following possibilities:

$N1$ the steady-state value Y may fail to exist;

$N2$ when Y is well-defined, generally $Y = S\{U, u(t)\}$ is a function of both U and $u(t)$;

$N3$ the dynamic response $y(t) = D\{U, u(t)\}$ is a function of both U and $u(t)$.

The first of these possibilities (i.e., $N1$) is illustrated by the first-order threshold autoregressive model considered in [70]:

$$y(k) = a|y(k-1)| + bu(k-1).$$

In particular, if the amplitude of the input to this model is sufficiently large, the step response of this model is *chaotic* and never settles out to a steady-state value Y. As an example of nonlinear behavior $N2$, consider the response of a Hammerstein model with the quadratic nonlinearity $g(x) = x^2$ to the input sequence $U+u(t)$ where $u(t) = \alpha cos(2\pi ft)$. It follows from simple trigonometric identities that, if K is the steady-state gain of the linear part of the Hammerstein model, then $Y = K(U+\alpha^2/2)$. That is, the steady-state response depends on both the steady-state input U and the amplitude of the "fluctuations" $u(t)$ about steady-state. Note that this "rectification phenomenon" — i.e., conversion of "fluctuations" into a "steady-state offset" — necessarily arises in any model with an asymmetric nonlinearity. This phenomenon is commonly observed in chemical processes; for a specific example, refer to the high-purity distillation column discussed in Section 9.

Finally, the nonlinear behavior $N3$ listed above is illustrated by the following example. Consider a Hammerstein model with the saturation nonlinearity:

$$g(x) = \begin{cases} x & |x| \leq 1 \\ 1 & |x| > 1, \end{cases}$$

and suppose the input to this model is $U + \alpha cos(2\pi ft)$ where $\alpha = 1/2$. For $-1/2 < U < 1/2$, this model behaves linearly and its dynamic response will be of the form $H(f)\alpha cos(2\pi ft)$, where $H(f)$ is the amplitude response of the linear part of the Hammerstein model at the excitation frequency f. For $-3/2 < U < -1/2$ and $1/2 < U < 3/2$, the saturation nonlinearity $g(x)$ will cause harmonic generation at frequencies $3f$, $5f$, ..., and the intensities of these harmonics will depend on U. Finally, for $U < -3/2$ or $U > 3/2$, the input nonlinearity will be fully saturated, so the input to the linear part of the Hammerstein model will be constant at either $+1$ or -1. Consequently, the output of the Hammerstein model will saturate at $\pm K$. This dependence of the dynamic response $y(t)$ on both U and $u(t)$ is also commonly observed in chemical processes and is again illustrated in Section 9.

The key point here is that the "steady-state behavior" of a nonlinear model is usually inextricably linked with both dynamic responses and the class of inputs under consideration. In addition, even when well-defined

steady-states exist for a given class of inputs, evaluating them may require significant effort. As a specific example, consider the task of characterizing the steady-state response of a NARMAX model of "moderate complexity" (e.g., 5 to 10 model parameters):

- does it exhibit chaotic regimes?
- are there pronounced differences in "steady-state gain" for different input sequence amplitudes?
- are there pronounced differences in "steady-state gain" for different input sequence "shapes"?
- etc.

Still, certain "steady-state characterizations" (e.g., "what does the step-response ultimately settle out to?") do yield useful insight into nonlinear model behavior and should be developed, if only as a prelude to a more complete dynamic characterization.

Development of a "Linear Skeleton"

Since the term "nonlinear models" automatically defines an alternative reference point — i.e., "linear models" — it is extremely useful to compare the performance of any nonlinear model with "nearby" linear ones. In addition, we may reverse this process, using linear models as a basis for guiding the development of nonlinear models. For example, in developing bilinear time-series models, Subba Rao and Gabr [92] advocate first constructing a linear model, taking advantage of such linear modeling tools as the Akaike Information Criterion (AIC) [14, 76, 96] to guide model order determination. Cinar [20] advocates a similar approach in developing NARMAX models: start with a "linear skeleton" that provides both guidance in selecting an initial model order and a basis for comparion in subsequent refinement steps. In addition, this approach may be taken a step further, "trimming" this "linear skeleton" through the use of techniques like stepwise regression to eliminate "marginal" model terms [92]; note that the "simplified" models developed by this procedure also serve as "secondary standards" for comparison in subsequent nonlinear model evaluations.

2.5 Nonlinear Model Structure Selection

As noted in Section 1, the first step in nonlinear process identification is the selection of an "appropriate" model structure. Unfortunately, this step is probably both the most difficult and the least amenable to systematic analysis. Consequently, this section offers brief summaries of the following four topics, to provide useful guidance:

1. parametric, nonparametric, and semi-parametric modeling;
2. variable selection and transformation;
3. local modeling;
4. control-motivated model structure selection.

2.5.1 Parametric, Nonparametric, and Semi-Parametric Models

Dynamic model identification is similar in some important respects to regression analysis. There, we are concerned with the identification of unknown functions (i.e., static maps) from available data. As a specific example, consider the scalar case: a real-valued function $f(\cdot)$ is to be identified such that the approximation $y \simeq f(x)$ is a "good" one for the available dataset $\mathcal{D} = \{(x_i, y_i)\}$. Three basic approaches have evolved for solving this problem — *parametric* regression, *nonparametric* regression, and *semi-parametric* regression. *Parametric* procedures postulate a general form for the function $f(\cdot)$, characterized by a p-vector of parameters θ. For example, an n^{th}-order scalar polynomial is characterized by $n+1$ coefficients. Given such a functional form, parametric procedures choose optimal values for these parameters by minimizing some measure of "lack of fit" (e.g., total squared error).

Nonparametric procedures do not assume an explicit form for the function, "allowing the data to choose." For example, kernel regression [42] constructs the following "smoothed" estimate of $f(x)$ from the available data:

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K(x - x_i) y_i}{\sum_{i=1}^{n} K(x - x_i)}. \tag{2.40}$$

Here, the function $K(\cdot)$ is a *smoothing kernel* that is typically of the form $K(x - x_i) = G(\frac{x - x_i}{h})$ where h is a constant parameter called the *bandwidth*. In practical terms, the end result of a nonparametric procedure is typically a

plot of the smooth function $\hat{f}(x)$. The choice of both the form of the "basic kernel function" $G(\cdot)$ and the bandwidth h are important in practice, but practical advice on these topics is becomming increasingly available [42, 58].

Semi-parametric regression procedures decompose the unknown function $f(x)$ into a parametric part and a nonparametric part, applying the appropriate analytical procedures to each. That is, we write $f(x) = f_1(\theta; x) + f_2(x)$ and develop a procedure that both estimates the unknown parameter vector θ and "smooths" the data appropriately to obtain an estimate $\hat{f}_2(x)$ of the "nonparametric part" $f_2(x)$.

These ideas carry over to system identification. For example, in linear system identification, direct identification of ARMAX models is a parametric problem. That is, autoregressive and moving-average model orders p and q are chosen and the corresponding model parameters $\{a_i\}$ and $\{b_i\}$ that best fit the available data are estimated. Alternatively, frequency responses may be estimated nonparametrically by first computing classical estimates of the input autospectrum $S_{uu}(f)$ and the input-output cross-spectrum $S_{uy}(f)$ [55, 76]. The ratio $\hat{H}(f) = S_{uy}(f)/S_{uu}(f)$ then yields a nonparametric estimate of the frequency response of the linear model relating the input/output data. The use of semi-parametric techniques in Hammerstein model identification is discussed briefly in Section 7.2.

The advantage of nonparametric estimates is that they impose fewer "prior assumptions" on the resulting model, but there are two significant disadvantages. First, note that nonparametric models are not generally applicable to control system design directly because most model-based control schemes require parametric models. For example, to be useful in a linear pole-placement procedure, a nonparametric frequency response estimate would first have to be converted to a rational transfer function model. Typically, this conversion would be accomplished by fitting a ratio of low-order polynomials to the estimated frequency response; the coefficients in this parametric model would then be used for controller design. In spite of this ultimate need for a parametric model, nonparametric procedures can be quite useful in the exploratory initial stages of model development. This is particularly true in nonlinear process identification because the model structure selection problem is so difficult, a point discussed further in Section 7.2. The second disadvantage of nonparametric approaches is that estimation of *multivariable* nonlinear functions rapidly becomes difficult [42]. In particular, it is not practical at present to apply a general nonparametric regression procedure to the NARMAX model (2.23) for "reasonable" values of p, q, and r and expect meaningful results. Thus, while nonparametric procedures

can be useful in the model structure selection process, they do not reduce the problem to a purely computational one.

2.5.2 Variable Selection and Transformation

Often, empirical process modeling leaves us a number of options regarding what variables to include in the model, and this choice can profoundly influence the nature of the model we ultimately develop. Specifically, if we have any discretion in the selection of variables to be included in a multivariable model, we may be able to influence the dynamic complexity, nonlinearity, and degree of interaction significantly, as the following simple example illustrates clearly. Consider a conical mixing tank with two feed streams — hot and cold water. The control objective is to maintain a fixed tank level and effluent temperature in the face of changes in effluent flow rate. Adopting a simple pairing scheme (e.g., controlling level by manipulating cold water flow rate, controlling temperature by manipulating hot water flow rate) leads to a nonlinear MIMO problem. If instead, we control tank level by manipulating total flow (i.e., the sum of hot and cold water flow rates) and control effluent temperature by manipulating an appropriately weighted difference between these flow rates, we obtain two weakly coupled SISO problems [66]. We can simplify this problem further by adopting the "extensive variable" control philosophy [34] and controlling not tank level, but the *volume* of material in the tank. That is, note that, given the height, minimum radius, and maximum radius of the tank, we can compute the volume of liquid in the tank from level measurements. The result of these changes in manipulated and controlled variables — based on physical insights — has taken us from a strongly nonlinear MIMO problem to two weakly coupled, weakly nonlinear SISO problems.

While this example is simple enough to be fairly obvious, it raises the point that the structure and complexity of an *empirical* model can depend dramatically on the exact choice of variables. Thus, if the physical variables available to us for empirical modeling are levels and individual flow rates, we may find that simpler models result if we consider such alternatives as the following:

- transform levels into approximate volumes;
- replace individual flow rates with total flow rates into and out of a vessel or "compartment";

- replace individual flow rates with ratios of individual flow rates;
- replace individual flow rates with ratios of individual flow rates to total flow rates.

Analytical procedures like nonparametric regression may also be useful in suggesting transformations to apply to individual variables to achieve simpler models. Finally, note that variable selection and transformations also influence robustness (i.e., sensitivity of model predictions to "outliers" or "bad data") [53].

2.5.3 Local Modeling

Another approach to the development of reduced complexity models is the development of multiple *local* models, each approximating process behavior "adequately" over some restricted operating range. The advantage of this approach is that the overall complexity of the collection of local models is often significantly less than that of the corresponding *global* model necessary to describe process dynamics over the entire operating range. For example, for many processes, if the range of control input variations is restricted sufficiently, it may be possible to approximate the process dynamics adequately by a linear model. Combining several linear models then leads to a model that is globally nonlinear but locally linear. This point is illustrated by Johansen and Foss [46] who develop a set of four local linear models to describe the dynamics of a batch fermentation reactor. The question of how *exactly* to specify the regime of validity of each model is an important one, as the following example illustrates.

Consider a single-input, single-output control problem and suppose the nonlinear process dynamics may be approximated "adequately" by a collection of m local linear models. Assume each model has a transfer function $H_i(s)$ and is associated with a set S_i of possible input values (e.g., intervals $[u_i^-, u_i^+]$). Define the m nonlinear functions:

$$G_i(u) = u\,\chi_i(u), \tag{2.41}$$

where $\chi_i(\cdot)$ is the *characteristic function* for the set S_i, i.e.:

$$\chi_i(u) = \begin{cases} 1 & u \in S_i \\ 0 & u \notin S_i. \end{cases} \tag{2.42}$$

Suppose we now construct an m-channel Uryson model (i.e., a PNL_m model in Chen's notation [17]) with the nonlinearity $G_i(\cdot)$ and the transfer function $H_i(s)$ in the i^{th} channel for $i = 1, 2, ..., m$. If the model validity sets S_i are disjoint, then for any value of the input u, only one of the "intermediate variables" $G_i(u)$ can be non-zero, so the Uryson model output will be the response of the "appropriate" linear model.

This construction suggests that for Uryson models in general, the quantity $\nu_i(u) = G_i(u)/u$ may be viewed as a "model validity function" for the linear model $H_i(s)$. That is, if $\nu_i(u)$ — a measure of the nonlinearity of $G_i(\cdot)$ — approximates the characteristic function for some set S_i of possible input values, the "local linear modeling" interpretation holds. Alternatively, if the functions $\nu_i(u)$ exhibit broad peaks over some range of input values, it may be appropriate to interpret them as *fuzzy set* characteristic functions [29, 101], describing imprecise regions of model validity. At the other extreme, if $\nu_i(u) = 1$ for all i and all u, the Uryson model reduces to a linear model whose transfer function is $H(s) = \sum_{i=1}^{m} H_i(s)$. Intermediate between these limits, the Uryson model may be viewed as a collection of local linear models whose regions of validity — and influence — overlap strongly. While it is not equivalent to the nonlinear Uryson model, a useful frame of reference for this intermediate case is the local linear model:

$$\tilde{H}(s) = \sum_{i=1}^{m} \nu_i(u_0)\, H_i(s), \tag{2.43}$$

which should have approximately the same steady-state behavior for $u(t) \simeq u_0$.

The preceeding discussion illustrates the close connection between local linear modeling and Uryson models, but two significant points have not yet been addressed. The first is that the region of validity for a local linear model is often not directly expressible in terms of the input u alone. For example, in the fermentor models developed by Johansen and Foss, the fermentor is a batch process and the models describe the evolution of the batch from fixed initial conditions, so there is no "control input u." Instead, three different combinations of output variables and state variables for the fermentor are used to specify the region of model validity, based on a qualitative understanding of the evolution of the batch through different operating regimes. This observation suggests that, if the analogy between local linear models and block-oriented models is to hold for these more complex cases, the "validity functions" $\nu_i(u)$ must be replaced by multivariable functions $\nu_i(u, x_1, ..., x_r)$ of some set of r measurable "auxiliary variables." This

change may destroy the block-oriented structure of the model, but investigation of the possibilities may yield useful insights into both block-oriented models and local modeling ideas.

The second point overlooked in the Uryson model discussion given above is the effect of transients that occur when switching from one region of model validity to another. That is, in switching from $u(t) \in S_i$ to $u(t') \in S_j$, both linear models i and j will have non-zero responses during the transient, which will be summed and appear in the Uryson model output. The question of how to combine local models is an important one and is considered in [2]. In addition, it is useful to note that the threshold autoregressive models discussed by Tong [96] are locally linear models. The fact that the subharmonic generating model discussed in Section 3.2 is of this form clearly illustrates that the behavior of "locally linear models" is quite distinct from that of "linear models."

Finally, it is important to note that, while the examples discussed here used local *linear* models, the local models need not be linear in general. In particular, recall that the fading memory systems discussed in Section 2 were defined on a specified set K of input sequences and exhibited a unique steady state for all inputs within that set. Further, for particular sets K (i.e., sets of bounded, "slew-rate limited" inputs), these systems could be approximated uniformly by finite Volterra models. This observation suggests that local Volterra modeling around each steady state may be a reasonable approach for systems (e.g., reactors) that are known to exhibit multiple steady states. Alternatively, local NARMAX models may be developed for different operating regimes of a process, resulting in a model with a wider overall range of validity and simpler structure than a global NARMAX model that attempts to describe all regimes simultaneously.

2.5.4 Control-Motivated Model Structure Selection

The primary motivation for empirical model identification given in Section 1 was the incompatibility of first-principles models with many control approaches due to the generally great complexity of such models. This observation provides a significant motivation for considering the "inverse problem": choosing a model structure that leads to a tractable control problem. Historically, this idea was one of the strongest motivations for the development of linear process models. More recently, this idea was one of the motivations for considering Volterra models — the model structure leads naturally to a special "nonlinear IMC" control approach [27].

Similarly, Zhu and Seborg [102] considered the "modified Hammerstein model" discussed earlier because it greatly simplifies the solution of the Model Predictive Control (MPC) problem relative to the standard Hammerstein model. Specifically, for the MPC problem, we ultimately need to solve for the control input $u(k)$ required to bring the predicted process output $\hat{y}(k)$ to some target value y^*. For a polynomial Hammerstein model, if $v(k)$ is the output of the static nonlinearity, it is given by:

$$v(k) = \sum_{i=1}^{L} \gamma_i [u(k)]^i, \tag{2.44}$$

while the modified Hammerstein model considered by Zhu and Seborg replaces this relation with the *dynamic* nonlinearity:

$$v(k) = \gamma_1 u(k) + \sum_{i=2}^{L} \gamma_i [u(k-1)]^i. \tag{2.45}$$

In the absence of either nonlinearity, the linear MPC problem gives the value of $v(k)$ such that $\hat{y}(k) = y^*$. To convert this result into a control input for the nonlinear problem, we must compute $u(k)$ from this value of $v(k)$. Zhu and Seborg proposed the modified Hammerstein model of Eq. (2.45) because inversion of Eq. (2.44) is a polynomial root-finding problem. In contrast, the inversion of Eq. (2.45) is accomplished by the simple analytical result:

$$u(k) = \frac{1}{\gamma_1} \left[v(k) - \sum_{i=2}^{L} \gamma_i [u(k-1)]^i \right], \tag{2.46}$$

which is directly computable since $u(k-1)$ is known at time k.

It was noted in Section 3 that this modified Hammerstein model is a member of the PPOD family of models. This observation raises a number of interesting questions. First and foremost is the question of how the class of dynamic nonlinearities described by this model differs from the range of behavior representable by the Hammerstein model. Zhu and Seborg's motivation for introducing it was the observation that if the sequence $\{u(k)\}$ is slowly varying, the control value $u(k)$ given by Eq. (2.46) should not differ too much from the solution to the unmodified Hammerstein problem. The more intriguing question is how the dynamics differ in cases where the difference between $u(k)$ and $u(k-1)$ is not small. In addition, referring back to the discussion of nonparametric approaches to Hammerstein modeling,

note that Zhu and Seborg's model generalizes directly from Eq. (2.45) to:

$$v(k) = \gamma_1 u(k) + g(u(k-1)), \tag{2.47}$$

where $g(\cdot)$ is an arbitrary function. In particular, nonparametric estimation of this function might expand the applicability of this model considerably, as in the unmodified Hammerstein case (see Section 7.2 for a discussion of this point). Finally, note that the MPC problem would remain tractable for a still wider class of PPOD models. For example, consider the following "control affine" model:

$$v(k) = \gamma_1 u(k)\, g(u(k-1)), \tag{2.48}$$

where again, $g(\cdot)$ is an arbitrary function. As in the previous examples, this structure permits explicit computation of $u(k)$ from $v(k)$ as:

$$u(k) = \frac{v(k)}{\gamma_1\, g(u(k-1))}. \tag{2.49}$$

An important question raised by such control-motivated model structures is how well they can represent the process to be controlled. In the examples just described, the general answer is "at least as well as the linear sub-class they contain." That is, viewing this linear sub-class as a constrained form of the nonlinear model, the best unconstrained model within the class must represent the process dynamics at least as well as the best constrained (i.e., linear) model within the class. In general, we would expect the best unconstrained model to be a better representation of the process dynamics, but the practical question is "how much better?"

2.6 Statistical Considerations

Once we have selected a model structure, the remaining three steps in the empirical model development sequence discussed in Section 1 all have strong connections with statistics. For this reason, the following section summarizes some important statistical concepts that will be useful in subsequent discussions of these modeling steps. This section is necessarily brief and emphasizes concepts, restricting consideration to discrete-time stochastic processes. Probably the most important notion considered here is the distinction between the distribution of a sequence of random variables and its dependence structure, both of which are important in nonlinear model identification. For

a more rigorous introduction to random variables and stochastic processes, see Billingsley [8]; for a more extensive "engineering-oriented" introduction, see Papoulis [68].

2.6.1 Working Definitions and General Concepts

The ideas presented here are mainly summarized from Rosenblatt [83], and rephrased slightly to emphasize the underlying concepts. The discrete-time stochastic processes considered here may be viewed as sequences $\{x_k\}$ of random variables where the index sequence k runs from $-\infty$ to $+\infty$. These sequences will be characterized in terms of the *joint densities* $p(x_j, x_{j+1}, ..., x_{j+m})$ for arbitrary subsequences $\{x_j, x_{j+1}, ..., x_{j+m}\}$. In particular, the following definitions are important.

The sequence $\{x_k\}$ is *statistically independent* if, for any j and m, the joint density $p(x_j, x_{j+1}, ..., x_{j+m})$ may be expressed as:

$$p(x_j, x_{j+1}, ..., x_{j+m}) = \prod_{i=0}^{m} p_{j+i}(x_{j+i}). \tag{2.50}$$

Here, the scalar function $p_{j+i}(\cdot)$ is the *marginal density* for the random variable x_{j+i}. A sequence that is *not* statistically independent will be called *dependent* and we will be very much concerned with its *dependence structure*, which is closely related to the ideas of dynamic modeling.

The sequence $\{x_k\}$ is *stationary* if, for any j and m, the joint density $p(x_j, x_{j+1}, ..., x_{j+m})$ is independent of j — i.e., if:

$$p(x_j, x_{j+1}, ..., x_{j+m}) = p(x_k, x_{k+1}, ..., x_{k+m}), \tag{2.51}$$

for arbitrary $k \neq j$. It follows that if $\{x_k\}$ is both statistically independent and stationary, the joint density is given by:

$$p(x_j, x_{j+1}, ..., x_{j+m}) = \prod_{i=0}^{m} p_0(x_{j+i}). \tag{2.52}$$

Here, all elements of the sequence $\{x_k\}$ are characterized by the single marginal density $p_0(\cdot)$, independent of the index k. Such sequences are called *independent, identically distributed* (i.i.d.) sequences, or "white noise" and are extremely important in the system identification literature.

Many standard results in statistics were first developed for i.i.d. sequences and then extended to sequences with "sufficiently weak" dependence structures. Commonly, any one of various "mixing conditions" are

considered [83], which are somewhat analogous to the fading memory conditions discussed in Section 2. Specifically, these mixing conditions imply that the sequence $\{x_k\}$ is "asymptotically independent," so that, for example, $p(x_k, x_j) \simeq p(x_k)p(x_j)$ if $|j-k|$ is sufficiently large. A particularly strong mixing condition is the assumption of "m-dependence," which implies that $p(x_k, x_j) = p(x_k)p(x_j)$ *exactly* if $|j-k| > m$. The connection between both of these conditions and NARMAX models will be discussed below.

2.6.2 Gaussian Sequences and Linear Models

If the joint densities $p(x_j, x_{j+1}, ..., x_{j+m})$ are *Gaussian* for all j and m, then the sequence $\{x_k\}$ is called a *Gaussian stochastic process.* A characteristic feature of Gaussian stochastic processes is that they are completely characterized by their first two moments: the means $\mu_k = E\{x_k\}$ and the autocorrelations $R_{xx}(j,k) = E\{x_k x_j\}$. For a *stationary* Gaussian stochastic process, the means are constant — i.e., $\mu_k = \mu$ for all k — and the autocorrelations depend only on the distance $|j-k|$ between the elements of the sequence, i.e.:

$$R_{xx}(j,k) \;=\; R_{xx}(|j-k|). \tag{2.53}$$

Similarly, the *autocovariance function* [68] or *second cumulant* [63] for a stationary stochastic process is defined as:

$$c_2^x(\tau) \;=\; E\{[x(k)-\mu][x(k+\tau)-\mu]\}. \tag{2.54}$$

Equivalently, a stationary Gaussian stochastic process may be completely characterized by its mean μ and its *power spectral density* $S_{xx}(f)$, defined as the discrete Fourier transform of the autocorrelation function, i.e.:

$$S_{xx}(f) \;=\; \sum_{k=-\infty}^{+\infty} e^{-j2\pi kfT} R_{xx}(k). \tag{2.55}$$

Here, T is the time between successive samples of the sequence $\{x(k)\}$ and f is a frequency variable that ranges from $-1/2T$ to $+1/2T$.

Probably the best known example of a Gaussian stochastic process is "Gaussian white noise," an i.i.d. sequence $\{x(k)\}$ of zero-mean, Gaussian random variables. The autocorrelation function for this sequence is given by:

$$R_{xx}(\tau) \;=\; \begin{cases} \sigma^2 & \tau = 0 \\ 0 & \tau \neq 0, \end{cases} \tag{2.56}$$