2 Modeling

2.1 Introduction

Signal analysis is frequently used to characterize systems. In van Drongelen (2007), chapter 8, we described linear systems and associated techniques that allow us determine system characteristics. In the last chapter of van Drongelen (2007) (section 17.3) we showed how these linear methods, such as cross-correlation, fail to characterize signals with a nonlinear component. To address this shortcoming, we used metrics such as correlation dimension, the Lyapunov exponent, or Kolmogorov entropy to characterize nonlinear signal properties.

The goal of this chapter is to introduce basics for modeling systems, with an emphasis on techniques used to characterize nonlinear systems and their signals. In this context, this chapter will also provide an introduction to the application of the Volterra series, which forms the basis for the identification of dynamical nonlinear systems, and which we will go over in more detail in Chapter 3. The systems that we will introduce in this chapter are considered higher-order systems, since they include operators beyond the (linear) first-order one. Useful references on the characterization of nonlinear systems are the seminal text by Marmarelis and Marmarelis (1978) and the reprint edition of a text from the 1980s by Schetzen (2006). For more recent overviews, see Westwick and Kearney (2003) and Marmarelis (2004).

2.2 Different Types of Models

Before going into mathematical detail, it is useful to summarize some of the types of models that one may encounter in neuroscience. Attenuators and amplifiers are both examples of **linear** systems, since output is simply the product of input and a constant (e.g., $y = 3x$). Alternatively, expressions that characterize **nonlinear** systems include higher-order terms: these systems, as we will see in Chapters 3–5, do not obey the scaling and superposition rules of linear models (to review these rules see section 8.3.1.1 in van Drongelen, 2007). Examples of nonlinear higher-order systems are $y = x^2$ (second-order system) and $y = 5 + x + 3x^3$ (third-order system). At this point it is important to note that an expression such as $y = a + bx + cx^3$ can still be considered linear, but with respect to its parameters a, b, and c. This is a

Signal Processing for Neuroscientists, A Companion Volume. DOI: 10.1016/B978-0-12-384915-1.00002-4

property that we will use when developing the regression procedure in Section 2.4.1.

All of the examples earlier are **static** models (systems without memory), meaning that their output depends only on present input. In neuroscience we usually must deal with **dynamical** models, in which output depends on present and past input (but not on future input); these systems are also called **causal**. Static models are represented by algebraic equations (such as the ones in the previous paragraph), whereas dynamical systems are modeled by differential equations (for continuous time models) or difference equations (for discrete time models). General examples of linear dynamical systems with input x and output y are:

$$A_n \frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n} + A_{n-1} \frac{\mathrm{d}^{n-1} y(t)}{\mathrm{d}t^{n-1}} + \cdots + A_0 y(t)$$
$$= B_m \frac{\mathrm{d}^m x(t)}{\mathrm{d}t^m} + B_{m-1} \frac{\mathrm{d}^{m-1} x(t)}{\mathrm{d}t^{m-1}} + \cdots + B_0 x(t)$$

for continuous time systems and:

$$A_n y(k-n) + A_{n-1} y(k-n+1) + \cdots + A_0 y(k)$$
$$= B_m x(k-m) + B_{m-1} x(k-m+1) + \cdots + B_0 x(k)$$

for discrete time systems (for details see chapter 8 in van Drongelen, 2007). If one of the terms in a differential or difference equation is of a higher order, we have a nonlinear dynamical system. For example, $y - 4(\mathrm{d}y/\mathrm{d}t)^2 = 2x$ represents a second-order dynamical system.

Time invariance is a critical condition for the development of the convolution formalism (see section 8.3.1.1 in van Drongelen, 2007). This property allows us to state that a system's response to identical stimuli at different points in time is always the same (provided that the system is in the same state, of course). Just as we have linear time invariant systems, we also have nonlinear time invariant systems (usually abbreviated as LTI or NLTI systems).

Models of real systems can be generated according to two major methodological approaches. One might follow a **deductive** path and start from (physical) assumptions about the system, generating a **hypothetical model** to create predictions that can be empirically tested. These empirical measurements can be used to establish the parameters of the hypothetical model, and therefore this type of representation is often called a **parametric model**. An alternative to this procedure, the **inductive** path, is followed if one starts from the measurements of a system's input and output. This data-driven method uses measurements, rather than assumptions about the system, to mathematically relate input and output. Here, we can consider the system as a **black box**, modeled by a mathematical relationship that transforms input into output. This type of model is often referred to as **nonparametric** (note, however, that nonparametric does not refer to the absence of parameters; in many cases,

these models will have more parameters than parametric models). The method of induction is appropriate when dealing with complex systems that resist a reduction to a simpler parametric model. It can also be a starting point in which a system is first characterized as a black box and in subsequent steps parts with physical meaning replace pieces of the black box. In this combined approach, parts of the model may still be part of the black box, whereas other parts may be associated with a physical interpretation. In this case, the distinction between parametric and nonparametric models may become a bit fuzzy.

2.3 Examples of Parametric and Nonparametric Models

A **parametric** model usually has relatively few parameters. A simple example of a parametric model of a dynamical LTI system is the ordinary differential equation (ODE) for a filter. For example, $x = RC(\mathrm{d}y/\mathrm{d}t) + y$ describes input x and output y of a simple RC circuit (Fig. 2.1A). The only parameters in this case are the values of the resistor R and the capacitor C in the equation. Subsequently, the value for these parameters can be determined experimentally from observing the system's behavior.

> *Note*: See van Drongelen (2007) for further details about determining these parameters from measurements: in section 11.2.1 it is shown how RC can be obtained from the filter's unit step response (equation (11.8)), and in section 12.3, RC is determined from the -3 dB point of the filter's frequency characteristic (equation (12.5)).

A very famous parametric model in neuroscience is the Hodgkin and Huxley (1952) formalism using four variables to describe the action potential generated in the squid's giant axon: the membrane potential V and three other variables m, h, and n describe the membrane potential-dependent characteristics of sodium and potassium conductivity. In the following it is assumed you are somewhat familiar with the Hodgkin and Huxley model; if you need to review the details, chapter 2 in Izhikevich (2007) provides an excellent overview.

Initially in the 1950s, the formalism was entirely hypothetical, and it was not until after the molecular basis for Na^+ and K^+ ion channels was elucidated that a physical interpretation of large parts of the model could be made. The gating variable m characterizes the depolarization process caused by increased conductance of sodium ions (causing an influx of positively charged sodium ions) that occurs during the action potential generation. The variables h and n are recovery variables that represent the inactivation of sodium ion conductance (reduced Na^+ influx) and the activation of potassium conductance (causing an outward flux of positively charged potassium ions).

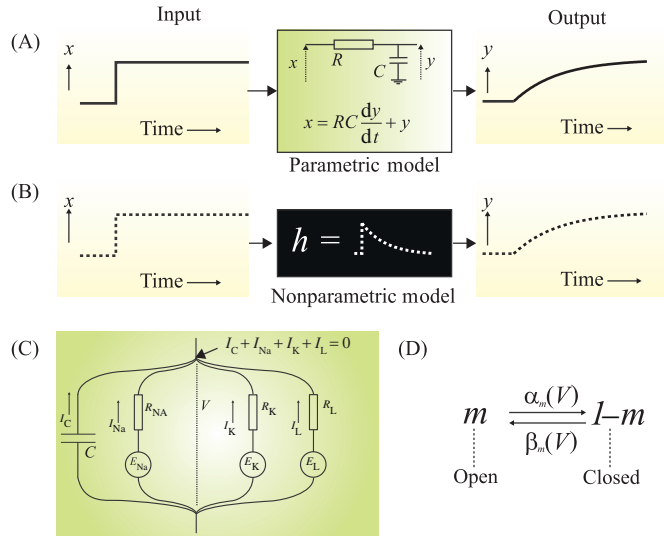

Figure 2.1 (A) Example of a parametric model of a dynamical linear system (a low-pass filter) and its input and output (x and y respectively). (B) The black box, nonparametric equivalent of the same system is the white curve representing the (sampled) unit impulse response (UIR). Both models permit us to predict the output resulting from an arbitrary input such as the unit step function. The parametric model has two parameters (R and C) with physical meaning. The nonparametric model consists of many parameters (the samples making up the UIR) without a direct physical meaning. (C) Hodgkin and Huxley's electronic equivalent circuit for the biomembrane. The model consists of the membrane capacitance (C) and three parallel ion channels: one for sodium, one for potassium, and a leakage channel. According to Kirchhoff's first law the sum of all currents at the node (*arrow*) must be zero. (D) Model for gating variable m in the Hodgkin and Huxley formalism.

Hodgkin and Huxley's model relates all these variables in an equivalent circuit of the excitable biomembrane (Fig. 2.1C) by setting the sum of all membrane currents equal to zero according to Kirchhoff's first law (see appendix 1.1 in van Drongelen, 2007). By applying this law to the node indicated with the arrow in the membrane model in Fig. 2.1C we obtain:

$$\underbrace{C\frac{dV}{dt}}_{\substack{\text{Capacitive}\\ \text{current}\\ =I_C}} + \underbrace{\frac{V-E_{Na}}{R_{Na}}}_{\substack{\text{Sodium}\\ \text{current}\\ =I_{Na}}} + \underbrace{\frac{V-E_K}{R_K}}_{\substack{\text{Potassium}\\ \text{current}\\ =I_K}} + \underbrace{I_L}_{\substack{\text{Leak}\\ \text{current}}} = 0 \qquad (2.1)$$

In this expression we have several parameters: membrane capacitance C; the resistance values for sodium and potassium ions, R_{Na} and R_K, respectively; E_{Na} and E_K

are the equilibrium potentials for sodium and potassium ions computed with the Nernst equation (appendix 1.1 in van Drongelen, 2007); and I_L is a constant leakage current attributed to Cl^- ions. The sodium and potassium currents are determined with Ohm's law (appendix 1.1 in van Drongelen, 2007): each ion species experiences a potential drop equal to the difference between the membrane potential V and its equilibrium potential (e.g., for sodium: $V - E_{Na}$), and this potential drop divided by the resistance is the ion current (e.g., for sodium the current is $(V - E_{Na})/R_{Na}$). In addition to Equation (2.1), Hodgkin and Huxley (1952) described the dynamics for R_{Na} and R_K with the nonlinear relationships $g_{Na} = 1/R_{Na} = \bar{g_{Na}} m^3 h$ and $g_K = 1/R_K = \bar{g_K} n^4$, where $\bar{g_{Na}}$ and $\bar{g_K}$ are the maximum conductivity values for sodium and potassium. Furthermore, the gating variable m is modeled by a reversible process between the open (m) and closed ($1-m$) states (Fig. 2.1D), which can be represented by the following ODE:

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \tag{2.2}$$

The rate parameters α_m and β_m that govern this process depend on the membrane potential V in a nonlinear fashion. The two other gating variables h and n follow the same formalism with membrane potential-dependent rate constants α_h, β_h, α_n, and β_n. Hodgkin and Huxley determined these nonlinear relationships between the rate parameters and membrane potential from voltage clamp experiments.

Over time, other ion channels were introduced into the model using the same formalism as for the sodium and potassium channels. Since the development of computer technology, the Hodgkin and Huxley formalism has been widely used in simulations of neural systems ranging from very detailed models of single neurons (e.g., De Schutter and Bower, 1994a, b) to large-scale networks of neocortex (e.g., Traub et al., 2005; van Drongelen et al., 2006).

Although Hodgkin and Huxley's model only contains four variables (V, m, h, n), it is still too complex to approach analytically. Several authors solved this problem by reducing the 4D model to a 2D one; the Fitzhugh–Nagumo model (Fitzhugh, 1961) is an example of such a reduction. In these models, the gating variable m of the Hodgkin and Huxley model is removed by considering sodium activation to be instantaneous; subsequently, h and n are combined into a single recovery variable w. Fitzhugh used the following pair of coupled differential equations:

$$\frac{dV}{dt} = V(a - V)(V - 1) - w + I \quad \text{and} \quad \frac{dw}{dt} = bV - cw \tag{2.3}$$

in which a, b, and c are parameters; I is a term representing injected current. The equations are coupled because w occurs in the expression for dV/dt and V in the expression for dw/dt. The remaining two variables in these models are the membrane potential V and a single recovery variable w, generating a 2D model that is amenable to mathematical analysis (for an excellent discussion of simplified versions of the Hodgkin and Huxley model, see Izhikevich, 2007).

Nonparametric models describe a system's input−output relationship, usually by using a large number of parameters, and these parameters do not necessarily have a physical interpretation. Generally speaking, a nonparametric model is generated from a procedure in which we relate a system's input $x(t)$ and output $y(t)$. Just as we can relate two variables with a function, we can link two time series with an **operator**. An example of such a nonparametric model would be the characterization of an LTI dynamical system with its (sampled) unit impulse response (UIR) (Fig. 2.1B). The operator in this case would be convolution, since convolution of the input time series $x(t)$ with the system's UIR $h(t)$ generates the system's output time series $y(t)$: $y(t) = h(t) \otimes x(t)$ (see section 8.3.1.1 in van Drongelen, 2007). Although one might point out that such a nonparametric description does not necessarily provide direct insight into the system's components or the mechanisms underlying the system's operation, the curve of the UIR permits us to predict the system's response to any input, such as the unit step function (Fig. 2.1B).

2.4 Polynomials

For static systems, both linear and nonlinear, one can use algebraic expressions to describe their input−output characteristic, and polynomials are often used for this purpose. Polynomials are sums of monomials, which are expressions that consist of a constant multiplied by one or more variables; the exponent of the variable is its degree. For example, $z(t) = ax(t)^4 y(t)^3$ is a monomial with a constant (parameter) a and a degree of 4 for x and 3 for y. We can see that this expression represents a static process because at any time t, output z depends only on the present values of inputs x and y. It is important to note that although the relationship between z and x, y is nonlinear, the expression can be considered a linear function of the parameter a.

2.4.1 Describing Discrete Time Data Sets

Applying the above to the characterization of nonlinear systems, we could describe the relationship between input $x(t)$ and output $y(t)$ of a static nonlinearity (a nonlinear system without memory) with a polynomial such as the following power series:

$$y(t) = a_0 + a_1 x(t) + a_2 x(t)^2 + a_3 x(t)^3 + \cdots + a_i x(t)^i + \cdots = \sum_{i=0}^{\infty} a_i x(t)^i \qquad (2.4)$$

In principle, power series are infinite; however, in our applications they will always consist of a finite number of monomials. The fact that Equation (2.4) **is linear with respect to its parameters a_i** can be used to fit the series by using a technique called **least squares** minimization. Using this approach of fitting polynomials to recorded data sets is often called **regression analysis**. This procedure works as follows. Suppose we have two sets of N measurements: a system's input x_n and associated output y_n. If we model our system as a second-order static system, we can

truncate the expression in Equation (2.4) above the second power and estimate the output y as $a_0 + a_1 x_n + a_2 x_n^2$. Subsequently we can define the error of our estimate ε^2 as:

$$\varepsilon^2 = \sum_{n=1}^{N} [y_n - (a_0 + a_1 x_n + a_2 x_n^2)]^2 \tag{2.5}$$

By following the same approach we used to find the coefficients in Lomb's algorithm (Section 1.2.1), we can find the minimum associated with the best choice for parameters a_0, a_1, and a_2 by setting the partial derivatives of ε^2 (with respect to these three parameters a_0, a_1, and a_2) equal to zero:

$$\frac{\partial \varepsilon^2}{\partial a_i} = \sum_{n=1}^{N} 2\left[y_n - (a_0 + a_1 x_n + a_2 x_n^2)\right] \frac{\partial [y_n - (a_0 + a_1 x_n + a_2 x_n^2)]}{\partial a_i} = 0 \quad \text{for } i = 0, 1, 2 \tag{2.6a}$$

and we get what are called the **normal equations**:

$$\begin{aligned}
\frac{\partial \varepsilon^2}{\partial a_0} &= -2\sum_{n=1}^{N} [y_n - a_0 - a_1 x_n - a_2 x_n^2] = 0 \\
&\rightarrow a_0 \underbrace{\sum_{n=1}^{N} 1}_{N} + a_1 \sum_{n=1}^{N} x_n + a_2 \sum_{n=1}^{N} x_n^2 = \sum_{n=1}^{N} y_n \\
\frac{\partial \varepsilon^2}{\partial a_1} &= -2\sum_{n=1}^{N} [y_n - a_0 - a_1 x_n - a_2 x_n^2] x_n = 0 \\
&\rightarrow a_0 \sum_{n=1}^{N} x_n + a_1 \sum_{n=1}^{N} x_n^2 + a_2 \sum_{n=1}^{N} x_n^3 = \sum_{n=1}^{N} y_n x_n \\
\frac{\partial \varepsilon^2}{\partial a_2} &= -2\sum_{n=1}^{N} [y_n - a_0 - a_1 x_n - a_2 x_n^2] x_n^2 = 0 \\
&\rightarrow a_0 \sum_{n=1}^{N} x_n^2 + a_1 \sum_{n=1}^{N} x_n^3 + a_2 \sum_{n=1}^{N} x_n^4 = \sum_{n=1}^{N} y_n x_n^2
\end{aligned} \tag{2.6b}$$

Note that in Equation (2.6b) all summation (Σ) expressions are numbers that can be computed from the observations; therefore, there are three linear equations with three unknown parameters a_0, a_1, and a_2 to compute (this should be no problem provided, of course, that the set of equations can be solved). Note that if we had truncated Equation (2.4) at a_1, the normal equations that we would have obtained would have been the well-known equations for linear regression.

It is a bit tedious to solve the three equations in (2.6b); therefore, one might prefer to solve the coefficients by using the matrix notation $XA = Y$ for the three equations:

$$\underbrace{\begin{bmatrix} N & \sum_{n=1}^{N} x_n & \sum_{n=1}^{N} x_n^2 \\ \sum_{n=1}^{N} x_n & \sum_{n=1}^{N} x_n^2 & \sum_{n=1}^{N} x_n^3 \\ \sum_{n=1}^{N} x_n^2 & \sum_{n=1}^{N} x_n^3 & \sum_{n=1}^{N} x_n^4 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \sum_{n=1}^{N} y_n \\ \sum_{n=1}^{N} y_n x_n \\ \sum_{n=1}^{N} y_n x_n^2 \end{bmatrix}}_{Y} \tag{2.6c}$$

The coefficients can be found by solving (Equation (2.6c)): that is, $A = X^{-1}Y$. In MATLAB we can use the `\` operator to obtain this result: `A = X\Y`. An example (`Pr2_1.m`) for approximating an exponential function $y = e^x$ is available on http://www.elsevierdirect.com/companions/9780123849151.

2.4.2 Describing Analytic Functions

The previous example works with discrete time data such as a set of digitized recordings of a system's input and output. In other applications, one might deal with a parametric model and consequently have access to analytic functions that describe some nonlinear system under investigation (recall that an **analytic function** is smooth and differentiable and that this is not the same as the analytic signal we introduced for the Hilbert transform in Chapter 1). In this case, the so-called Maclaurin or Taylor series approaches, which will be explained in Sections 2.4.2.1 and 2.4.2.2, may be applied to convert the function into a power series. Such a power series approach can also be helpful for creating a linear approximation of a nonlinear function in the neighborhood of a point of interest. Because linear relationships are easier to analyze than nonlinear ones, this technique of linearization of nonlinear functions can help us understand the behavior of complex nonlinear processes.

Like the polynomials discussed in the previous section, the Maclaurin and Taylor series describe static systems. To describe dynamical systems, we can use the Volterra series, which is discussed in detail in Chapter 3. In Section 2.5, we will show that the Taylor series can be considered the static version of a Volterra series.

2.4.2.1 Maclaurin Series

A famous power series describing a function about the origin is the **Maclaurin** series. Let us consider an example with the exponential function and use the power series approach in Equation (2.4) to represent this function:

$$f(t) = e^t = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_i t^i + \cdots = \sum_{i=0}^{\infty} a_i t^i \tag{2.7}$$

The task at hand is to determine the values of the coefficients a_i for function e^t. We can use the following approach to perform this task. First we determine the derivatives of f.

$$\begin{aligned}
f(t) &= e^t = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_i t^i + \cdots \\
\frac{df(t)}{dt} &= e^t = a_1 + 2a_2 t + 3a_3 t^2 + \cdots + i a_i t^{i-1} + \cdots \\
\frac{d^2 f(t)}{dt^2} &= e^t = 2a_2 + (2 \times 3) a_3 t + \cdots + (i \times (i-1)) a_i t^{i-2} + \cdots \\
\frac{d^3 f(t)}{dt^3} &= e^t = (2 \times 3) a_3 + \cdots + (i \times (i-1) \times (i-2)) a_i t^{i-3} + \cdots \\
&\vdots
\end{aligned} \tag{2.8}$$

The second step is to consider $f(t) = e^t$ about the origin. As we approach the origin (i.e., t becomes 0), Equation (2.8) simplifies to:

$$\begin{aligned}
f(0) &= e^0 = 1 = [a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_i t^i + \cdots]_{t=0} = a_0 \\
\frac{df(0)}{dt} &= e^0 = 1 = [a_1 + 2a_2 t + 3a_3 t^2 + \cdots + i a_i t^{i-1} + \cdots]_{t=0} = a_1 \\
\frac{d^2 f(0)}{dt^2} &= e^0 = 1 = [2a_2 + (2 \times 3) a_3 t + \cdots + (i \times (i-1)) a_i t^{i-2} + \cdots]_{t=0} = 2a_2 \\
\frac{d^3 f(0)}{dt^3} &= e^0 = 1 = [(2 \times 3) a_3 + \cdots + (i \times (i-1) \times (i-2)) a_i t^{i-3} + \cdots]_{t=0} \\
&= (2 \times 3) a_3 \\
&\vdots
\end{aligned} \tag{2.9}$$

With the results obtained in Equation (2.9), we can see that for the function, the values for the coefficients a_i are:

$$a_i = \frac{1}{i!} \tag{2.10}$$

Combining this result in Equation (2.10) with Equation (2.7), we have found the well-known power series expansion of the exponential function:

$$\boxed{f(t) = e^t = 1 + \frac{1}{1!} t + \frac{1}{2!} t^2 + \frac{1}{3!} t^3 + \cdots + \frac{1}{i!} t^i + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!} t^i} \tag{2.11}$$

In the last expression $\sum_{i=0}^{\infty}(1/i!)\,t^i$, we use the definition $0! \equiv 1$. Note that by using this approach, we include only values of t—there are no previous or future values ($t \pm \tau$) included in the power series; therefore, this approach is **static** (or memoryless). An example of this approximation is implemented in MATLAB script Pr2_1.m (http://www.elsevierdirect.com/companions/9780123849151).

In the above example we used the exponential $\exp(t)$ for $f(t)$; if we consider the development of Equation (2.4) for any function that can be differentiated, we get:

$$\begin{aligned}
f(0) &= [a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_i t^i + \cdots]_{t=0} = a_0 \rightarrow a_0 = f(0) \\
\frac{\mathrm{d}f(0)}{\mathrm{d}t} &= [a_1 + 2a_2 t + 3a_3 t^2 + \cdots + i a_i t^{i-1} + \cdots]_{t=0} = a_1 \rightarrow a_1 = f'(0) \\
\frac{\mathrm{d}^2 f(0)}{\mathrm{d}t^2} &= [2a_2 + (2\times 3)a_3 t + \cdots + (i\times(i-1))a_i t^{i-2} + \cdots]_{t=0} = 2a_2 \rightarrow a_2 = \frac{f''(0)}{2} \\
\frac{\mathrm{d}^3 f(0)}{\mathrm{d}t^3} &= [(2\times 3)a_3 + \cdots + (i\times(i-1)\times(i-2))a_i t^{i-3} + \cdots]_{t=0} \\
&= (2\times 3)a_3 \rightarrow a_3 = \frac{f'''(0)}{(2\times 3)} \\
&\vdots
\end{aligned} \tag{2.12}$$

Here the notation $f'(0)$, $f''(0)$, $f'''(0)$, ... are not functions but represent the numbers computed as the value of the 1st, 2nd, 3rd, ... derivatives of f at $t = 0$. From this more general notation we obtain the expression for the so-called Maclaurin series of $f(t)$:

$$\boxed{f(t) = f(0) + \frac{1}{1!}t f'(0) + \frac{1}{2!}t^2 f''(0) + \frac{1}{3!}t^3 f'''(0) + \cdots} \tag{2.13}$$

2.4.2.2 Taylor Series

In the above example, we developed the power series for a function about the origin. The development of the Taylor series follows a similar approach but now about any point α. For a power series of power N this becomes:

$$\begin{aligned}
f(t) &= a_0 + a_1(t-\alpha) + a_2(t-\alpha)^2 + a_3(t-\alpha)^3 + \cdots + a_i(t-\alpha)^i + \cdots \\
&= \sum_{i=0}^{\infty} a_i (t-\alpha)^i
\end{aligned} \tag{2.14}$$

We will now use a similar approach for the development of this series about α as we used in the Maclaurin series about the origin—except in this case we set $t = \alpha$

(instead of $t=0$) so that all terms in Equation (2.14) with $(t-\alpha)^i$ vanish. By following this procedure we get:

$$\begin{aligned}
f(\alpha) &= [a_0 + a_1(t-\alpha) + a_2(t-\alpha)^2 + a_3(t-\alpha)^3 + \cdots + a_i(t-\alpha)^i + \cdots]_{t=\alpha} \\
&= a_0 \rightarrow a_0 = f(\alpha) \\
\frac{\mathrm{d}f(\alpha)}{\mathrm{d}t} &= [a_1 + 2a_2(t-\alpha) + 3a_3(t-\alpha)^2 + \cdots + ia_i(t-\alpha)^{i-1} + \cdots]_{t=\alpha} \\
&= a_1 \rightarrow a_1 = f'(\alpha) \\
\frac{\mathrm{d}^2 f(\alpha)}{\mathrm{d}t^2} &= [2a_2 + (2\times 3)a_3(t-\alpha) + \cdots + (i\times(i-1))a_i(t-\alpha)^{i-2} + \cdots]_{t=\alpha} \\
&= 2a_2 \rightarrow a_2 = \frac{f''(\alpha)}{2} \\
\frac{\mathrm{d}^3 f(\alpha)}{\mathrm{d}t^3} &= [(2\times 3)a_3 + \cdots + (i\times(i-1)\times(i-2))a_i(t-\alpha)^{i-3} + \cdots]_{t=0} \\
&= (2\times 3)a_3 \rightarrow a_3 = \frac{f'''(\alpha)}{(2\times 3)} \\
&\vdots
\end{aligned} \tag{2.15}$$

Similar to the notation used in the previous section, the notation $f'(\alpha)$, $f''(\alpha)$, $f'''(\alpha)$, ... does not refer to functions, but represents the numbers computed as the value of the 1st, 2nd, 3rd, ... derivatives of f at $t=\alpha$. Substituting the findings in Equation (2.15) into Equation (2.14) we obtain the Taylor series about $t=\alpha$:

$$\boxed{f(t) = f(\alpha) + \frac{1}{1!}(t-\alpha)f'(\alpha) + \frac{1}{2!}(t-\alpha)^2 f''(\alpha) + \frac{1}{3!}(t-\alpha)^3 f'''(\alpha) + \cdots} \tag{2.16}$$

Comparing Equations (2.13) and (2.16), we can establish that the Maclaurin series is the same as a Taylor series computed about the origin (i.e., $\alpha = 0$). This approach can be extended to higher-dimensional systems with multiple inputs; see Appendix 2.1 for examples of the 2D case. It must be noted that in many texts the distinction between Maclaurin and Taylor series is not always made and it is not uncommon to use the term Taylor series for both, a habit we will adopt in the following.

The number of terms in a Taylor series may be infinite. However, if we evaluate a system close to an equilibrium at the origin or α, the value of t or $(t-\alpha)$ is a small number $\ll 1$; therefore, one can ignore higher-order terms in the power series t^n or $(t-\alpha)^n$ because they become increasingly smaller. Thus, in general we can approximate any function close to α with a linear expression obtained from a Taylor series in which higher-order terms are ignored $f(t) \approx f(\alpha) + (t-\alpha)f'(\alpha)$; or, in the case where we evaluate the expression about the origin, we obtain the approximation $f(t) \approx f(0) + tf'(0)$. This technique of **linearizing a nonlinear function** plays an important role in the analysis of nonlinear systems. A system's

behavior in a restricted part of its domain can be understood and approximated by a linear version of its characteristic. Sometimes, with the more complex systems, a piecewise approximation with linear functions is the best option for their analysis. For example, if we wanted to evaluate $\sin(t)$ around the origin, we can apply Equation (2.13) and find the series:

$$\sin(t) = \underbrace{\sin(0)}_{0} + t\underbrace{\cos(0)}_{1} - \frac{t^2}{2}\underbrace{\sin(0)}_{0} - \frac{t^3}{6}\underbrace{\cos(0)}_{1} + \cdots$$

For small values of t (around 0) we may ignore all higher-order terms and we find that $\sin(t) \approx t$. In general, such an approach may be useful if one studies a system close to an equilibrium. For example, if one examines a neuron's subthreshold behavior, one must describe the membrane potential close to the resting potential; in this case it makes sense to linearize the nonlinear equations that govern the cell's electrical activity around resting potential. An example of this approach, where the nonlinear Hodgkin and Huxley equations are linearized, can be found in Chapter 10 in Koch (1999).

When fitting a truncated power series to an analytic function, one could truncate the Taylor series at the desired order. However, due to the error introduced by truncation, one may actually obtain a better fit by using a linear regression approach. An example is if one wants to approximate e^t with a second-order power function over a limited interval. The truncated Taylor series (see Equation (2.11)) is $1 + t + 0.5t^2$, but with a regression approach over the interval $[-1,1]$ one obtains a better fit with $0.9963 + 1.1037t + 0.5368t^2$. This can be seen by running MATLAB script Pr2_1 (available on http://www.elsevierdirect.com/companions/9780123849151) where the original exponential function (red), the Taylor series (blue), and the regression result (black) are superimposed. The regression approach for obtaining a power series approximation is also a valid solution if the Taylor series cannot be applied, as in the case of a function that is nonanalytic, such as $y = |x|$ (no [unique] derivative at $x = 0$).

2.5 Nonlinear Systems with Memory

In the above examples, the output $y(t)$ of the nonlinear systems could be described with a polynomial of $x(t)$ because there was a direct relationship between x and y; that is, in these examples there was no memory in the system. However, nonlinear systems with memory do exist, and for these systems we must describe how the output $y(t)$ depends on both the present and the past input: $x(t-\tau)$ with $\tau \geq 0$.

In the following chapter, we will consider the details of the so-called Volterra series for the characterization of dynamical nonlinear systems (nonlinear systems that do have a memory). Here we will demonstrate the similarities between the Volterra and Taylor series. With the Taylor series we can link output value $y = f(x)$ to input value x in the following manner:

$$y = f(\alpha) + \frac{1}{1!}(x-\alpha)f'(\alpha) + \frac{1}{2!}(x-\alpha)^2 f''(\alpha) + \frac{1}{3!}(x-\alpha)^3 f'''(\alpha) + \cdots \qquad (2.17)$$

In the example below, we will approximate a nonlinearity with a series truncated at the second order:

$$y(t) = a_0 + a_1 x(t) + a_2 x(t)^2 \tag{2.18}$$

Before we introduce the Volterra series, we generalize the procedure in which we relate two **values** x and y into a slightly altered procedure in which we relate a pair of **time series** $x(t)$ and $y(t)$. Just as we can relate two values x and y with a **function** f:

$$y = f(x) \tag{2.19a}$$

we can link two time series $x(t)$ and $y(t)$ with an **operator** F:

$$y(t) = F\{x(t)\} \tag{2.19b}$$

> *Note*: In some texts on Volterra series F will be called a **functional**. Because F connects two functions $x(t)$ and $y(t)$, it is better to use the term "operator" because strictly speaking, a **functional** maps a function onto a value, whereas an **operator** maps one function to another function.

A Volterra series can perform such an operation:

$$\begin{aligned} y(t) = & \underbrace{h_0}_{\text{0th order term}} + \underbrace{\int_{-\infty}^{\infty} h_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1}_{\text{1st order term}} \\ & + \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}_{\text{2nd order term}} + \cdots \\ & + \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_n(\tau_1,\tau_2,\cdots,\tau_n)x(t-\tau_1)x(t-\tau_2)\cdots x(t-\tau_n)\mathrm{d}\tau_1\,\mathrm{d}\tau_2\cdots\mathrm{d}\tau_n}_{n\text{th order term}} \end{aligned} \tag{2.20}$$

Do not be intimidated by this first appearance of the expression for the Volterra series. In the following text and Chapter 3 we will discuss and explain the different components of this representation. At this point it is worthwhile to mention that the Volterra series is essentially the convolution integral extended to nonlinear systems. We could simplify the notation in Equation (2.20) with the commonly used symbol for convolution $\otimes$ (chapter 8 in van Drongelen, 2007), and we get:

$$y = h_0 + h_1 \otimes x + h_2 \otimes x \otimes x + \cdots + h_n \underbrace{\otimes x \otimes \cdots \otimes x}_{n \text{ copies of } x}$$

In the Volterra series (Equation 2.20), input function $x(t)$ determines the output function $y(t)$. The expression is analogous to the Taylor series except that the differentials of the Taylor series are replaced by integrals. The symbols h_0, h_1, h_2, and h_n represent the so-called Volterra kernels. The term "kernel" is uniquely defined for this type of series and should not be confused with the use of this term in computer science or other areas in mathematics. Note that the first-order component $\int_{-\infty}^{\infty} h_1(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1$ in the Volterra series is the convolution integral (see section 8.3.1.1 in van Drongelen, 2007) and the higher-order components in Equation (2.20) are convolution-like integrals. Thus for a linear system, kernel h_1 is the UIR. Representations that utilize Volterra series are usually nonparametric—that is, one can predict system output when the input is known, but one cannot necessarily intuit the system's components or underlying mechanisms. In the following we will examine examples of the relationship between Volterra and Taylor series. See also Chapter 3 for further details on the Volterra series.

Despite the similarities between the Taylor series in Equation (2.17) and the Volterra series in Equation (2.20) discussed above, it may not be immediately obvious that they are related. Therefore, we will discuss the similarities for a simple dynamical nonlinear system, which we will subsequently transform into a static nonlinear one. Let us consider a dynamical second-order system that consists of a cascade of a dynamical linear component and a static nonlinear module (Fig. 2.2A). Such a cascade approach with the dynamics in the linear component combined with static nonlinearities is frequently applied in dynamical nonlinear system analysis. In this example, we have the linear component's UIR $h(t)$ and the static second-order nonlinear component $a_0 + a_1 y + a_2 y^2$ (Equation (2.18)). From Fig. 2.2A we can establish that the output y of the linear module can be obtained from the convolution of the input x and the linear module's UIR h:

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau \tag{2.21}$$

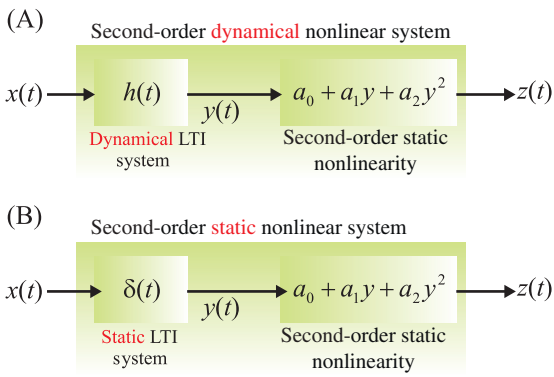

Figure 2.2 (A) Diagram of a second-order dynamical nonlinear system consisting of a cascade of a dynamical LTI system and a second-order static nonlinearity. (B) A similar system for which the dynamical linear component is replaced by a static one.

The cascade's final output z can be obtained from the static nonlinearity characteristic by substituting the output of the linear component (Equation (2.21)) into the input of the static nonlinearity (Equation (2.18)):

$$z(t) = a_0 + a_1 \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau + a_2 \left[\int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau \right]^2 \tag{2.22}$$

This can be rewritten as:

$$\begin{aligned} z(t) = a_0 &+ a_1 \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau \\ &+ a_2 \underbrace{\left[\left(\int_{-\infty}^{\infty} h(\tau_1)x(t-\tau_1)\mathrm{d}\tau_1 \right) \left(\int_{-\infty}^{\infty} h(\tau_2)x(t-\tau_2)\mathrm{d}\tau_2 \right) \right]}_{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\tau_1)h(\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2} \end{aligned} \tag{2.23}$$

This expression can be rearranged in the form of the Volterra series shown in Equation (2.20):

$$\begin{aligned} z(t) = &\overbrace{\underbrace{a_0}_{h_0}}^{\text{0th order term}} + \overbrace{\int_{-\infty}^{\infty} \underbrace{a_1 h(\tau)}_{h_1(\tau)} x(t-\tau)\mathrm{d}\tau}^{\text{1st order term}} \\ &+ \overbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{a_2 h(\tau_1)h(\tau_2)}_{h_2(\tau_1,\tau_2)} x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}^{\text{2nd order term}} \end{aligned} \tag{2.24}$$

Equation (2.24) shows that the system in Fig. 2.2A can be characterized by a Volterra series for a second-order system with Volterra kernels h_0, h_1, and h_2.

To demonstrate that the Taylor series is the static equivalent of the Volterra series, we show the equivalence of Equation (2.24) to the power series in Equation (2.18). To accomplish this, we consider the case where our dynamical component in the cascade becomes static; the linear component is now replaced by the static function $y(t) = x(t)$. In other words, the linear module's UIR is the unit impulse δ

itself, indicating that for this linear component output equals input (Fig. 2.2B). Therefore, we can substitute $\delta(t)$ for $h(t)$ in Equation (2.24):

$$\begin{aligned} z(t) &= a_0 + a_1 \underbrace{\int_{-\infty}^{\infty} \delta(\tau)x(t-\tau)\mathrm{d}\tau}_{x(t)} \\ &\quad + a_2 \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\tau_1)\delta(\tau_2)x(t-\tau_1)x(t-\tau_2)\mathrm{d}\tau_1\,\mathrm{d}\tau_2}_{x(t)x(t)=x(t)^2} \\ &= a_0 + a_1 x(t) + a_2 x(t)^2 \end{aligned} \tag{2.25}$$

Thus, in the static case, we can use the Volterra series to recover $z(t) = a_0 + a_1 x(t) + a_2 x(t)^2$, which is the original expression of the power series in Equation (2.18).

> *Note*: The integrals in Equation (2.25) are evaluated using the property of the unit impulse δ (see section 2.2.2 in van Drongelen, 2007): $\int_{-\infty}^{\infty} x(\tau)\delta(\tau)\mathrm{d}\tau = x(0)$ and accordingly $\int_{-\infty}^{\infty} x(t-\tau)\delta(\tau)\mathrm{d}\tau = x(t)$.

Appendix 2.1

Taylor Series for a 2D Function

We can extend the Taylor series in Equation (2.16) to a function $f(\tau, \sigma)$ of two variables τ and σ. In the case where we can subdivide the function into two separate ones (e.g., $f(\tau,\sigma) = f(\tau) + f(\sigma)$ or $f(\tau,\sigma) = f(\tau)f(\sigma)$), we can compute the Taylor series for each function $f(\tau)$ and $f(\sigma)$ and add or multiply the individual series to obtain the expression for $f(\tau, \sigma)$. Such an approach would work if, for example, $f(\tau,\sigma) = \mathrm{e}^{\tau}\sin(\sigma)$.

Alternatively, one can approach the development of a 2D Taylor series more generally, and consider f about point α, β.

$$\begin{aligned} f(\tau,\sigma) &= a_{00} + a_{10}(\tau-\alpha) + a_{01}(\sigma-\beta) + a_{20}(\tau-\alpha)^2 + a_{11}(\tau-\alpha)(\sigma-\beta) \\ &\quad + a_{02}(\sigma-\beta)^2 + a_{30}(\tau-\alpha)^3 + a_{21}(\tau-\alpha)^2(\sigma-\beta) \\ &\quad + a_{12}(\tau-\alpha)(\sigma-\beta)^2 + a_{03}(\sigma-\beta)^3 + a_{40}(\tau-\alpha)^4 + \cdots \end{aligned} \tag{A2.1.1}$$

Using a similar approach as the one for the single-variable Taylor series, we set τ and σ to α and β and find $f(\alpha, \beta) = a_{00}$. To find the other coefficients we use partial differentiation of f at point α, β:

$$\frac{\partial f(\alpha,\beta)}{\partial \tau} = a_{10}, \quad \frac{\partial f(\alpha,\beta)}{\partial \sigma} = a_{01}, \quad \frac{\partial^2 f(\alpha,\beta)}{\partial \tau^2} = 2a_{20},$$
$$\frac{\partial^2 f(\alpha,\beta)}{\partial \tau\, \partial \sigma} = a_{11}, \quad \frac{\partial^2 f(\alpha,\beta)}{\partial \sigma^2} = 2a_{02} \tag{A2.1.2}$$

This technique can be used to obtain the full power series of f. In most applications we are interested in the linear approximation of the 2D series:

$$\boxed{f(\tau,\sigma) \approx f(\alpha,\beta) + \frac{\partial f(\alpha,\beta)}{\partial \tau}(\tau - \alpha) + \frac{\partial f(\alpha,\beta)}{\partial \sigma}(\sigma - \beta)} \tag{A2.1.3a}$$

The higher-order nonlinear terms are often not considered because we assume that we only look at f closely around point α, β; therefore, $\tau - \alpha$ and $\sigma - \beta$ are very small numbers, and higher powers of these small contributions are even smaller. In other words, when f is in the neighborhood of point α, β, the function can be approximated with the linear terms in Equation (A2.13a). In many cases, especially in physics literature, you may encounter an alternative notation for the linear approximation of a nonlinear system. The small fluctuations $\tau - \alpha$ and $\sigma - \beta$ around α, β are indicated as perturbations $\delta\tau$ and $\delta\sigma$, and the notation for $f(\alpha, \beta)$, $(\partial f(\alpha, \beta))/\partial\tau$, and $(\partial f(\alpha, \beta))/\partial\sigma$ is changed to $[f]_{\alpha,\beta}$, $[\partial f/\partial\tau]_{\alpha,\beta}$, and $[\partial f/\partial\sigma]_{\alpha,\beta}$:

$$\boxed{f(\tau,\sigma) \approx [f]_{\alpha,\beta} + \left[\frac{\partial f}{\partial \tau}\right]_{\alpha,\beta} \delta\tau + \left[\frac{\partial f}{\partial \sigma}\right]_{\alpha,\beta} \delta\sigma} \tag{A2.1.3b}$$

Again, recall that in this notation $[f]_{\alpha,\beta}$, $[\partial f/\partial\tau]_{\alpha,\beta}$, and $[\partial f/\partial\sigma]_{\alpha,\beta}$ represent the coefficients in the equation. They are numbers and not functions, since these represent the function and its derivatives when evaluated at point α, β. An example of an application that linearizes the nonlinear Hodgkin and Huxley equations can be found in Chapter 10 of Koch (1999).