theory offers tools better suited than computational theory to understanding biological systems as fundamentally temporal, dynamic systems operating in changing, uncertain environments. For this reason, we adopt modern control theory as a means of understanding neurobiological systems.

After a brief introduction to modern control theory, we spend the remainder of the chapter developing successively more involved biological examples. These examples show how control theory is a natural ally of the account of representation and transformation developed so far.

8.1 CONTROL THEORY AND NEURAL MODELS

8.1.1 Introduction to control theory

There are two equations that effectively summarize linear control theory:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{8.1}$$

$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t). \tag{8.2}$$

These equations constitute what is known as the *internal description* of a linear system. Together, they are often called the *state equations* of a system because the vector $x(t)$ consists of variables that together describe the (internal) state of the system. It is the dynamics of these variables that is of the most interest. This is because the *state vector*, $x(t)$, serves to summarize the effects of all past input. Thus, all future output depends only on the current value of the state vector and the future input. In a sense, then, the state vector is a bridge between the past input and the future output.

The remaining variables in these equations are understood as follows (see figure 8.1): $u(t)$ is the input or control vector, $y(t)$ is the output vector, $\mathbf{A}$ is the dynamics matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output or sensor matrix, and $\mathbf{D}$ is the feedthrough matrix. The dimensions of each of the vectors can be different, but these dimensions determine the corresponding matrix dimensions. Thus, there can be multiple inputs, outputs, and state variables, whose interactions are determined by the elements of the matrices.

The system depicted in figure 8.1 is called *time-invariant* because none of the matrices are functions of time. This means that the parameters controlling the dynamics of the system are taken not to change over the period of interest. This is often a useful simplifying assumption, but it is clearly unrealistic in describing neurobiological systems. We discuss both time-invariant and time-varying systems (see section 8.2). One assumption that we do not relax is that the systems we are interested in are *lumped* systems. A lumped system is one whose state variables are finite. A system whose state has infinitely many variables
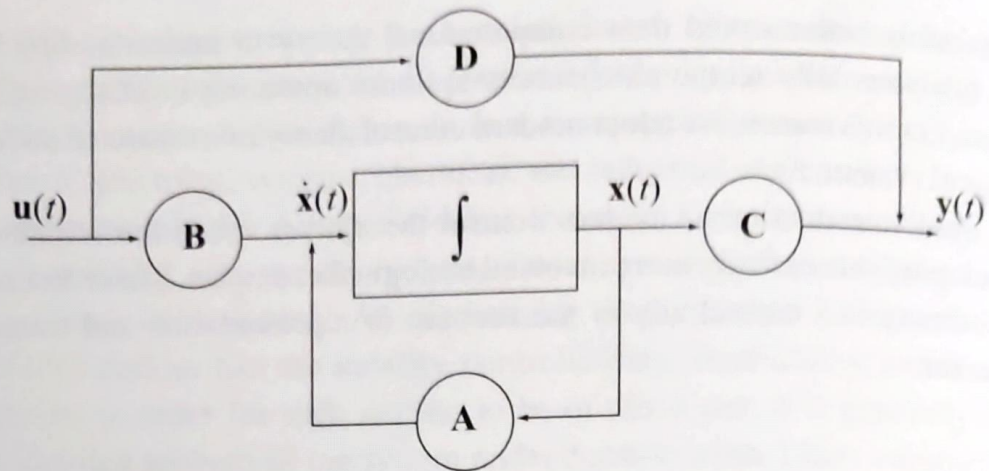
Figure 8.1
A generic block diagram for a time-invariant linear system showing the elements of equations (8.1) and (8.2). In this diagram $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input or control vector, $\mathbf{y}(t)$ is the output vector, $\mathbf{A}$ is the dynamics matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output or sensor matrix, and $\mathbf{D}$ is the feedthrough matrix.

is called a *distributed* system. Of course, any distributed system can be arbitrarily well approximated by a sufficiently large lumped system. The examples we deal with are all reasonably small lumped systems.

8.1.2   A control theoretic description of neural populations

Given the ubiquity of nonlinear transformations in neurobiological systems, it is far from clear that such linear analyses will be of use. This, in fact, is not a problem unique to neurobiological systems. Because there is no general theory for analyzing nonlinear systems, the study of such systems is generally based on linear systems theory. We adopt this same strategy. So, while we consider both linear and nonlinear neurobiological systems, we always base our analyses on linear control theory.

The means by which we integrate control theory and our previous analyses of representation and transformation is captured by the third principle of neural engineering: neural dynamics can be characterized by taking neural representations to be control theoretic state variables. In other words, elements of the state vector, x, (i.e., the state variables) *just are* the representations we find in neural populations. And, the relations between these variables (as defined by the system matrices) *are* the transformations implemented by these populations. However, we cannot take the control theoretic descriptions of the relations between variables at face value. Rather, we need to relate the intrinsic dynamics of neural populations to the standard control theoretic characterization. This allows for a 'translation' between standard control theory and a neurally relevant system description.

In figure 8.1, there are four matrices and a transfer function. In order for this to be a dynamic system, the transfer function (integration in this figure) needs to be a function of

time. Given how we have so far characterized neural function, there are two possibilities for transfer functions in a neural population; that is, there are two parts of the system that are inherently dynamic. One is the synaptic filter, $h(t)$, and the other is the soma voltage, $V(t)$. Surprisingly, we can show that the synaptic filter generally dominates the dynamics of the cellular response as a whole (see appendix F.1).

As mentioned earlier, a good approximation to synaptic dynamics is given by (see section 4.4.2)[2]

$$h(t) = \frac{1}{\tau} e^{-t/\tau}, \tag{8.3}$$

where τ is the synaptic time constant. This function describes the postsynaptic current (PSC) produced by the arrival of a spike and, of course, it is the linear decoder we discussed extensively in chapter 4. So, we need to characterize the dynamics of neural systems with respect to this decoder.

We can begin by drawing a 'neural' control diagram (see figure 8.2), like that in figure 8.1. This diagram is missing the feedthrough and output matrices from figure 8.1. However, in the case of a neural system, where each such subsystem describes a single population, both the feedthrough matrix and the output matrix can be taken to be incorporated into the input matrix of a subsequent population. Thus, these two matrices need not be explicitly considered here. Note also that the original integration is replaced by $h(s)$, the Laplace transform of the synaptic dynamics, $h(t)$, in equation (8.3).

It is very convenient to use the Laplace transform when characterizing dynamics with differential equations. Essentially, this transform provides a means of writing and manipulating differential equations algebraically. Much like the Fourier transform, the independent variable for the Laplace transform, s, can be thought of as frequency. The following equation defines the Laplace transform:

$$\begin{aligned} \mathcal{L}(f(t)) &= \int_0^\infty e^{-st} f(t)dt \\ &= f(s). \end{aligned}$$

Here, $f(t)$ is in the time domain and $f(s)$ is in the frequency domain.

Taking the Laplace transform of the time-invariant internal description, equations (8.1) and (8.2), gives

---

2 This is a significant simplification, though a good approximation to true PSCs. In particular, it ignores the rapid (yet finite) rise in the observed PSC. We have used various forms for this filter in numerical experiments and have found that including a finite rise time tends to improve the dynamics of the simulation. However, the increased complexity of analyzing more faithful models serves to obscure the insights that can be gained by (at least initially) considering a simpler model. Nevertheless, similar techniques can be employed to analyze other synaptic models.
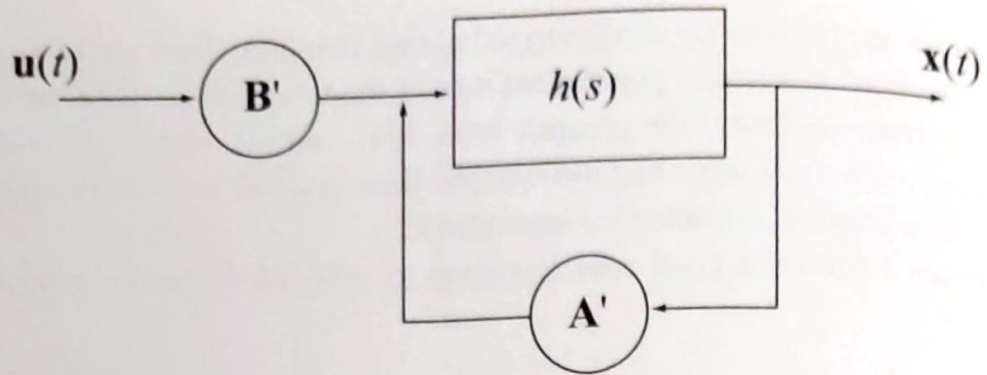
Figure 8.2
A generic neural population as a linear system. Note that the feedthrough matrix and output matrix are not included as part of the figure because both can be incorporated into the input matrix of the subsequent population. $\mathbf{A}'$ and $\mathbf{B}'$ are the *neural* dynamics and *neural* input matrices respectively. The transfer function, $h(s)$, is the Laplace transform of the synaptic dynamics.

$$s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s) \tag{8.4}$$

$$\mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s). \tag{8.5}$$

Given figure 8.2, we can see that equation (8.5) will always be $\mathbf{y}(s) = \mathbf{x}(s)$ for a neural subsystem, so we focus on equation (8.4). The system described by figure 8.2 can be written as

$$\mathbf{x}(t) = h(t) * [\mathbf{A}'\mathbf{x}(t) + \mathbf{B}'\mathbf{u}(t)]$$

or, in the frequency domain,

$$\mathbf{x}(s) = h(s)[\mathbf{A}'\mathbf{x}(s) + \mathbf{B}'\mathbf{u}(s)]. \tag{8.6}$$

The Laplace transform, $h(s)$, of $h(t)$ as given by (8.3) is

$$h(s) = \frac{1}{1 + s\tau}.$$

Therefore,

$$\mathbf{x}(s) = \frac{1}{1 + s\tau}[\mathbf{A}'\mathbf{x}(s) + \mathbf{B}'\mathbf{u}(s)]$$

$$= \frac{\tau^{-1}}{\tau^{-1} + s}[\mathbf{A}'\mathbf{x}(s) + \mathbf{B}'\mathbf{u}(s)].$$

So,

$$(\tau^{-1} + s)\mathbf{x}(s) = \tau^{-1}[\mathbf{A}'\mathbf{x}(s) + \mathbf{B}'\mathbf{u}(s)]$$

$$s\mathbf{x}(s) = \tau^{-1}[\mathbf{A}' - \mathbf{I}]\mathbf{x}(s) + \tau^{-1}\mathbf{B}'\mathbf{u}(s). \tag{8.7}$$

Equating the right-hand sides of (8.4) and (8.7) defines the relation between the dynamics and input matrices for the standard control system and the neural control system. Namely,

$$\mathbf{A}' = \tau\mathbf{A} + \mathbf{I} \tag{8.8}$$

$$\mathbf{B}' = \tau\mathbf{B}, \tag{8.9}$$

where $\mathbf{I}$ is the appropriately dimensioned identity matrix.

As simple as these equations are, given a control system in the standard form of figure 8.1, we can use them to determine the equivalent neural system in the standard form of figure 8.2. We present a detailed example of this in section 8.2, where we re-examine the neural integrator previously presented in sections 2.3 and 5.3.

8.1.3 Revisiting levels of analysis

Before examining specific examples that use this relation, let us consider how the preceding analysis can help articulate the theory as it has been presented so far. Recall that in chapter 2 we introduced the distinction between basic representations (i.e., neural activities) and higher-level representations (i.e., encoded physical properties). In this section we examine how higher-level and basic representations can be related using the control theoretic approach just discussed. As well, we show how to move easily between descriptions employing these different levels of representation. As a result, we characterize a generic neural subsystem that can be combined to construct large, complex models with various degrees of neural realism (possibly within the same model).

To understand the subsequent discussion, a few comments on notation are in order. First, we adopt the conventions from linear control theory that, in systems diagrams: 1) boxes denote transfer functions (which define the dynamics of the system); 2) circles denote matrices that multiply their inputs; and 3) intersecting lines indicate addition. We also introduce a convention to use a rightward pointing triangle to indicate a function of the input that may be nonlinear. As well, we use Greek superscripts to index *populations*, and Roman subscripts to index *neurons* within a population.

Based on the discussion in the previous section, we can draw a standard subsystem at the basic level (see figure 8.3). Here we have a system that is consistent with typical neuroscientific descriptions of neuron function. Spike trains, $\sum_n \delta^\beta(t - t_{jn})$, from various preceding populations, β, arrive at the synaptic cleft and, by the action of neurotransmitters, induce a current change, $\omega_{ij}^{\alpha\beta} h(t)$, in the postsynaptic dendrites of neurons in population α. These resulting postsynaptic currents (PSCs) are filtered versions of the presynaptic spike train, where the filter is well-approximated by $1/1 + s\tau_{ij}^{\alpha\beta}$. Either at the time of filtering or due to various dendritic processes, these PSCs can have varying degrees of effect on the changes in the somatic current, $J_i^\alpha(t)$. The synaptic weights, $\omega_{ij}^{\alpha\beta}$, model this effect. There