Appendix B

A Miniprimer on Linear Systems Analysis

For those readers who forgot linear systems analysis, crucial to this book, we here provide the briefest of reviews.

A system is always linear or nonlinear with respect to some particular input and output variable. Indeed, the same physical system can be linear when using one sort of input-output pairing and nonlinear when considering a different one. If we restrict ourselves to the case when the input and output variables are single-valued functions of time, termed \( x(t) \) and \( y(t) \), respectively, then for a system \( L \), defined as

\[
y(t) = L[x(t)]
\]

(B.1)
to be linear, it must obey two constraints. Firstly, it must be homogeneous, that is,

\[
L[\alpha x(t)] = \alpha L[x(t)].
\]

(B.2)

For instance, doubling the input should double the output. Secondly, the system must also be additive,

\[
L[x_1(t) + x_2(t)] = L[x_1(t)] + L[x_2(t)].
\]

(B.3)

The response of the system to the sum of two inputs is given by the sum of the responses to the individual inputs. These two properties are sometimes also summarized in the superposition principle, expressed as

\[
L[\alpha x_1(t) + \alpha_2 x_2(t)] = \alpha_1 L[x_1(t)] + \alpha_2 L[x_2(t)].
\]

(B.4)

A further property that some (not all) linear systems possess is shift or time invariance; in other words, if the input is delayed by some \( \Delta t \), the output will be delayed by the same interval. A linear system is shift invariant if and only if

\[
y(t) = L[x(t)] \quad \text{implies} \quad y(t - t_1) = L[x(t - t_1)].
\]

(B.5)

If a system possesses these three properties, then its entire behavior can be summarized by its response to an impulse or delta function \( \delta(t) \).\(^1\) The impulse response or Green's function of the system is exactly what its name implies, namely, the response of the system to an impulse,

\[
h(t) = L[\delta(t)].
\]

(B.6)

Any input signal can be treated as an infinite sum of appropriately shifted and scaled impulses, or

\[
x(t) = \int_{-\infty}^{+\infty} x(t_1) \delta(t - t_1) dt_1.
\]

(B.7)

The properties of homogeneity, additivity, and shift invariance ensure that the response to any arbitrary input \( x(t) \) can be obtained by summing over appropriately shifted and scaled responses to an impulse function (or, equivalently, the output is a weighted sum of its inputs). In short,

\[
y(t) = \int_{-\infty}^{+\infty} x(t_1) h(t - t_1) dt_1.
\]

(B.8)

The shorthand form of this integral operation, known as a convolution, is \(*\), as in

\[
y(t) = (x * h)(t).
\]

(B.9)

We conclude that once we know \( h(t) \), the response of a linear system to an impulse, the response to an arbitrary input waveform can be obtained by linear convolution operation.

Before we end this short digression, we briefly want to remind the reader of another way to analyze time-invariant linear systems, namely, by using sinusoidal inputs. If the input to a linear system is a sinusoidal wave of a particular frequency \( f \) (in hertz), the output is another sinusoidal of the same frequency but shifted in time and scaled,

\[
L[\sin(2\pi ft)] = A(f) \sin(2\pi ft + \phi(f)).
\]

(B.10)

The function \( A(f) \) is known as the amplitude response and determines how much the output is scaled for an input at frequency \( f \), while the phase \( \phi(f) \) determines how much the sinusoidal wave at the output is shifted in time with respect to the input.

Any input can always be represented as a sum of shifted and scaled sinusoids. For a linear system, the impulse response function and the amplitude and phase functions are closely related by way of the Fourier transform. The Fourier transform of the impulse response function \( h(t) \) is

\[
\tilde{h}(f) = F[h(t)] = \int_{-\infty}^{+\infty} e^{-i2\pi ft} h(t) dt.
\]

(B.11)

The amplitude of this filter \( \tilde{h}(f) \) corresponds to the amplitude of the Fourier transform of the impulse response function. Note that \( \tilde{h}(f) \) (throughout the book, the \( \tilde{h} \) symbol denotes the Fourier transform of some function \( h \)) is a complex function. We then have

\[
\tilde{h}(f) = A(f) e^{i\phi(f)}.
\]

(B.12)

The reason we frequently talk in this book about the input being "filtered" by the filter function \( \tilde{h}(f) \) is that formally the output can be obtained by convolving (another name for filtering) the input by the filter function. In the frequency space representation, convolution is turned into a straight multiplication, and Eq. B.8 can be rewritten as

\[
y(f) = \tilde{h}(f) \cdot \tilde{x}(f).
\]

(B.13)

As emphasized before, when discussing linearity it is crucial to discuss with respect to what variable. There are a number of instances in which neurobiological systems can be treated,
Appendix C

Sparse Matrix Methods for Modeling Single Neurons

Barak A. Pearlmutter and Anthony Zador

In this appendix we describe numerical methods used in the efficient solution of the linear and nonlinear cable equations that describe single neuron dynamics. Our exposition is limited to the scale of a whole neuron; we will ignore both the simulation of circuits of neurons (see, for example, the monographs by Bower and Beeman, 1998 and by Koch and Segev, 1998), as well as the simulation of the stochastic equations governing single ion channel kinetics (Skaugen and Walloe, 1979; Chow and White, 1996).

The appendix is divided into two parts. The first deals with the solution of the linear component of the cable equation. Since the cable as well as the diffusion equation are linear second-order parabolic partial differential equations (PDE), this part draws on techniques and principles developed in many other fields that deal with similar equations, though naturally the discussion will focus on those problems of particular interest in neurobiology. The theme common to this part is that for the purposes of numerical solution, the cable equation is best discretized into a system of ordinary differential equations coupled by sparse matrices. The main difficulty is that the resulting equations are stiff, that is, they display time scales of very different magnitude; even so it is possible to apply widely available and efficient techniques for sparse matrices. The second part deals with the nonlinear components of neurodynamics, particularly equations of the Hodgkin–Huxley type and those arriving from calcium dynamics. These nonlinearities are surprisingly benign, and can readily be handled with a few simple techniques, provided that the linear component is treated properly.

Many of the techniques described are implemented in widely used and freely available neural simulators (in particular GENESIS and NEURON; see DeSchutter, 1992; Hines, 1998; Bower and Beeman, 1998) in a manner that is relatively transparent to the user. Nevertheless, there are at least three good reasons for understanding the foundations of these numerical methods. Firstly, when such simulators produce surprising—and possibly spurious—results, an understanding of their internal workings can help determine whether the numerical method is to blame. Secondly, when conducting original research it is inevitable that some problem will arise for which the software must be customized. Finally, and most important, understanding these techniques provides insight into the underlying neurodynamics itself, and hence into the behavior of neurons.