

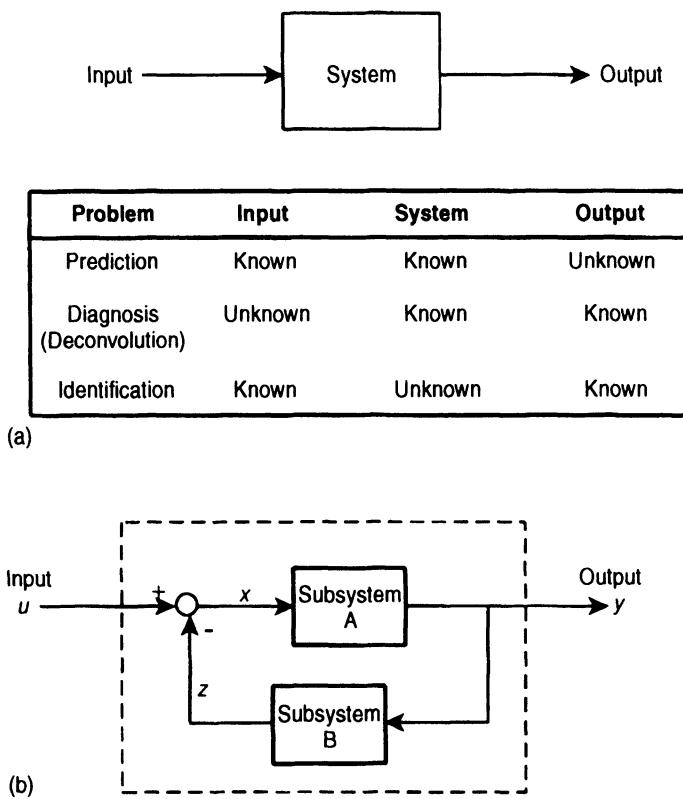
# 7

# Identification of Physiological Control Systems

## 7.1 BASIC PROBLEMS IN PHYSIOLOGICAL SYSTEM ANALYSIS

In the past several chapters, we have examined a variety of techniques for analyzing the steady-state and dynamic characteristics of feedback control systems. A common thread among all these different methods has been the use of the *systems approach*: the physiological process under study is decomposed into a number of interconnected “systems” or “subsystems,” under the assumption that each of these components can be characterized functionally by a set of differential equations or their Laplace equivalents. This is displayed schematically in Figure 7.1a. Up to this point, we have always assumed that the equations in each of the “boxes” are known or can be derived by applying physical principles and physiological insight to the process in question. Thus, knowing the form of the independent variable (the “input”), the equations (representing the “system”) can be solved to deduce the form of the dependent variable (the “output”). This type of analysis is known as the *forward problem* or *prediction problem*. Predictions allow us to determine whether the model postulated provides an accurate characterization of the process under study. A somewhat greater challenge is posed by the *inverse problem*. Here, a model of the process in question is available and the output is measured; however, the input is not observable and therefore has to be deduced. This is known as the *diagnosis problem* and often involves the need for *deconvolution* of the model impulse response with the output in order to deduce the input.

The third type of problem, that of *system identification*, is the most pervasive in physiological system analysis. There are two basic approaches to system identification. It is often the case, when dealing with physiological processes, that we have only very limited insight into the underlying mechanisms or that the complexity of the processes involved is just too overwhelming. Under such conditions, it would be difficult to begin with physical principles to derive the differential equations that appropriately characterize the system under study. At this level of knowledge (or lack of it), it probably would be more useful to probe the system in question with known stimuli and to record the system’s response to these inputs. This is the *black-box* or *nonparametric* approach to system identification, where little is



**Figure 7.1** (a) Three fundamental problems in system analysis. (b) Identification of a closed-loop system.

assumed about the system except, perhaps, whether we expect it to be linear or nonlinear. Ideally, we would be able to deduce, from the measured input and output, the system impulse response (if it is linear) or kernels (if it is nonlinear; see Marmarelis and Marmarelis, 1978) and use these to catalog the behavior of the unknown process. The result would be a purely *empirical* model of the system under study.

In the case of systems for which *some* knowledge regarding mechanisms is available, it is generally possible to put this knowledge to use by coming up with a mathematical description (which could consist of a set of differential or difference equations or their frequency-domain equivalents). This characterizes the second approach to system identification, in which a *structural* or *gray-box* model is constructed. While the “structure” of such a model is derived from what we know about the physiological process being studied, there remain unknown model coefficients or *parameters* that have to be determined. Thus, the next stage in system identification in such cases, following model-building, is the problem of *parameter estimation*. For this reason, structural models fall into the category referred to as *parametric models*. In the case of the linear lung mechanics model that we analyzed in Chapters 4 through 6, the unknown parameters ( $R$ ,  $L$  and  $C$ ) each bear a one-to-one correspondence to a physiological entity—airway resistance, fluid inertance, and lung compliance. But this is not a requirement of all parametric models. *Functional models* are models that contain only parameters that can be estimated from input–output data. Frequently, some of these parameters may be related to the underlying physiological entities but a one-to-

one correspondence may not exist. In many models of pharmacokinetics, for example, there are often assumed “compartments” that may be used to account for effects arising from many different sources, but not one single definable physiological entity. In some other functional models, a negative delay may have to be postulated; such a parameter clearly has no physiological meaning but may be needed in order to fully characterize the observed system behavior.

The control engineering literature is replete with countless methods of system identification, particularly for linear systems. In this chapter, we will discuss the few basic techniques that have been most commonly applied in physiological system analysis. While there is a large body of literature on the theory of system identification in simple single-input-single-output systems, there has been relatively much less work published on the identification of closed-loop systems. The fact that most physiological systems are closed-loop can introduce some complications into the process of system identification or parameter estimation. Referring to the example illustrated in Figure 7.1b, if we could only measure the input ( $u$ ) and the output ( $y$ ) of the overall closed-loop system, but the internal variables  $x$  and  $z$  were unobservable, it would be impossible for us to know (just based on the measurements of  $u$  and  $y$ ) that this is in fact a closed-loop system. On the other hand, if we could measure  $x$  in addition to  $u$  and  $y$ , we would in principle be able to identify subsystem A; then, having identified the overall closed-loop system, we would be able to determine subsystem B. Similarly, if we could measure the feedback variable  $z$ , it would be possible (at least in theory) to determine subsystem B; then, from knowledge of the overall closed-loop model, we would be able to deduce subsystem A. This is the fundamental basis of *closed-loop estimation*. In some cases, it may be possible to “open the loop,” through surgical, physiological, or pharmacological interventions. Indeed, some of the biggest advances in physiology have resulted from clever experimental designs that allowed the researchers to “open the loop” in one or more of these ways. We will review some examples of these in the sections to follow.

## 7.2 NONPARAMETRIC AND PARAMETRIC IDENTIFICATION METHODS

We begin by reviewing some of the basic computational techniques commonly employed in the identification of single-input-single-output, open-loop systems. While most physiological models have been developed assuming a continuous-time base, in practice physiological measurements are generally obtained as *discrete-time* samples of the signals under study. Some of these time series contain measurements obtained at a fixed sampling rate, e.g., arterial blood pressure and the electroencephalograph. However, some measurements that involve pulsatile or cyclic quantities, e.g., heart rate and respiratory rate, are sampled at a fixed phase of each cycle; these intervals are generally not equally spaced in time. Since the process of system identification requires the use of real data, the vast majority of identification techniques that have been developed assume a discrete-time base. Thus, for the most part, system identification problems are solved by numerical methods and do not have closed-form analytical solutions. An important assumption that we will make, however, is that the sampling interval has been selected to be small enough that the time series obtained adequately capture the fastest dynamics present in the observed signals. Sampling the input and output signals at rates that are lower than one-half of the highest frequency present in the signals can lead to the problem of *aliasing*, in which the sampled data may appear to contain dynamic components that were really not contained in the original signals. Use of these aliased input and output time series would definitely lead to erroneous estimates of the system impulse responses or transfer functions.

### 7.2.1 Numerical Deconvolution

The most direct nonparametric techniques for linear system identification have been discussed earlier in Chapter 4. The response to the step input has been one of the most commonly used methods for characterizing physiological system dynamics, provided the stimulus can indeed be made to follow a time-course that closely approximates a step. Having found the step response, the impulse response can be deduced by differentiating the former with respect to time. In general, impulsive inputs cannot easily be implemented in physiological applications. If the step input is also not a convenient option, one might resort to stimulating the system under study with a bolus type of input. Then, in order to estimate the impulse response from the bolus response and the input, one can employ the method of numerical deconvolution.

Assuming that the data samples are obtained at a uniformly spaced time interval,  $T$ , the convolution expression relating input ( $u$ ) and impulse response ( $h$ ) to output ( $y$ ) is represented in discrete time as:

$$y(nT) = \sum_{k=0}^n h(nT - kT)u(kT)T \quad (7.1)$$

where the current time  $t = nT$ . In the special cases where  $n = 0, 1$ , and  $2$ , Equation (7.1) becomes

$$y(0) = h(0)u(0)T \quad (7.1a)$$

$$y(T) = [h(T)u(0) + h(0)y(T)]T \quad (7.1b)$$

Thus, from

$$y(2T) = [h(2T)u(0) + h(T)u(T) + h(0)y(2T)]T \quad (7.1c)$$

Equation (7.1a), assuming  $u(0) \neq 0$ , we find that

$$h(0) = \frac{y(0)/T}{u(0)} \quad (7.2a)$$

Similarly, rearranging Equation (7.1b), we obtain

$$h(T) = \left\{ \frac{y(T)}{T} - u(T)h(0) \right\} / u(0) \quad (7.2b)$$

so that, once  $h(0)$  has been deduced from Equation (7.2a), the next point in the impulse response function,  $h(T)$ , can be determined from Equation (7.2b). Subsequently, from Equation (7.1c), we get

$$h(2T) = \left\{ \frac{y(2T)}{T} - h(T)u(T) - h(0)y(2T) \right\} / u(0) \quad (7.2c)$$

where  $h(2T)$  can be determined, since  $h(0)$  and  $h(T)$  are now known. This estimation procedure is continued for all subsequent values of  $h(t)$ . Thus, the general deconvolution formula is

$$h(nT) = \left\{ \frac{y(nT)}{T} - \sum_{k=1}^n h(nT - kT)u(kT) \right\} / u(0) \quad (7.2d)$$

While Equation (7.2d) is valid in principle, in practice it is hardly used. The reason is that small values for  $u(0)$  can amplify errors enormously, and errors made in each sequential estimate of  $h$  tend to accumulate.

### 7.2.2 Least Squares Estimation

A key problem of numerical deconvolution is that the estimated impulse response function is “forced” to satisfy Equation (7.1) even when it is clear that the output measurements,  $y$ , will contain noise. The effect of this noise accumulates with each step in the deconvolution process. One way to get around this problem is to build some averaging into the estimation procedure. To do this, we restate the problem in the following way: Given  $N$  pairs of input–output measurements, estimate the impulse response function (consisting of  $p$  points, where  $p \ll N$ ) that would allow Equation (7.1) to be satisfied *on average*. To develop this mathematically, we recast the relationship between input and output measurements in the following form:

$$y(nT) = \sum_{k=0}^{p-1} h(kT)u(nT - kT)T + e(nT), \quad n = 0, 1, \dots, N-1 \quad (7.3)$$

where  $e(nT)$  represents the error between the measured (noisy) value and the “best estimate” of the response at time  $t = nT$ . The “best estimate” of the response is obtained by selecting the impulse response function  $\{h(kT), k = 0, 1, \dots, p-1\}$  that would minimize the sum of the squares of all the errors,  $\{e(nT), n = 0, 1, \dots, N-1\}$ . Thus, this method is analogous to the fitting of a straight line to a given set of data-points, except that the “line” in this case is a multidimensional surface.

To find the least-squares estimate of  $h(t)$ , we proceed by defining the following matrix and vector quantities:

$$\underline{y} = [y(0) \ y(T) \ \dots \ y((N-1)T)]' \quad (7.4)$$

$$\underline{h} = [h(0) \ h(T) \ \dots \ h((N-1)T)]' \quad (7.5)$$

$$\underline{e} = [e(0) \ e(T) \ \dots \ e((N-1)T)]' \quad (7.6)$$

and

$$\mathbf{U} = \begin{bmatrix} u(0) & 0 & \dots & 0 \\ u(T) & u(0) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ u((N-1)T) & u((N-2)T) & \dots & u((N-p)T) \end{bmatrix}^T \quad (7.7)$$

In Equations (7.4) through (7.6),  $\underline{y}$ ,  $\underline{h}$  and  $\underline{e}$  are column vectors (and the prime ‘ represents the transpose operation). Then, the system of equations represented by Equation (7.3) can be compactly rewritten in matrix notation as

$$\underline{y} = \mathbf{U}\underline{h} + \underline{e} \quad (7.8)$$

Let  $J$  represent the sum of squares of the errors. Then,

$$J = \sum_{n=0}^{N-1} e(nT)^2 = \underline{e}'\underline{e} \quad (7.9)$$

Combining Equation (7.8) and Equation (7.9), we get

$$J = (\underline{y} - \mathbf{U}\underline{h})'(\underline{y} - \mathbf{U}\underline{h}) \quad (7.10)$$

To find the minimum  $J$ , we differentiate Equation (7.10) with respect to the vector  $\underline{h}$ , and equate all elements in the resulting vector to zero:

$$\frac{\partial J}{\partial \underline{h}} = -2\mathbf{U}'\underline{y} + 2\mathbf{U}'\mathbf{U}\underline{h} = \mathbf{0} \quad (7.11)$$

Rearranging Equation (7.11), we find that the least squares solution for the impulse response function is

$$\underline{h} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\underline{y} \quad (7.12)$$

It can be shown further that a lower bound to the estimate of the variance associated with the estimated elements of  $\underline{h}$  is given by

$$\text{var}(\underline{h}) = (\mathbf{U}'\mathbf{U})^{-1}\sigma_e^2 \quad (7.13)$$

where  $\sigma_e^2$  is the variance of the residual errors  $\{e(nT), n = 0, 1, \dots, N-1\}$ , i.e.,

$$\sigma_e^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} e(nT)^2 \quad (7.14)$$

This method produces much better results for  $h(t)$  than numerical deconvolution, since we are using  $N$  pieces of information to deduce estimates of  $p$  unknowns, where  $p$  should be substantially smaller than  $N$ . How small the ratio  $p/N$  should be depends on the relative magnitude of noise in the data. As a rough rule of thumb,  $p/N$  should be smaller than 1/3. Another requirement for obtaining good estimates of  $h(t)$  is that the matrix  $\mathbf{U}'\mathbf{U}$  must not be ill-conditioned since it has to be inverted: as one can see from Equation (7.13), the variance of  $h(t)$  becomes infinite if  $\mathbf{U}'\mathbf{U}$  is singular. Since  $\mathbf{U}$  consists of all the input measurements, the conditioning of  $\mathbf{U}'\mathbf{U}$  depends on the time-course of the stimulus sequence. This problem will be discussed further in Section 7.3.

The practical implementation of this method is relatively straightforward in MATLAB. An example of the MATLAB code that can be used to estimate the elements of vector  $\underline{h}$  and their associated standard errors (in column vector  $\underline{hse}$ ) is given in the script file “sysid\_ls.m.” The main portion of this code is displayed below.

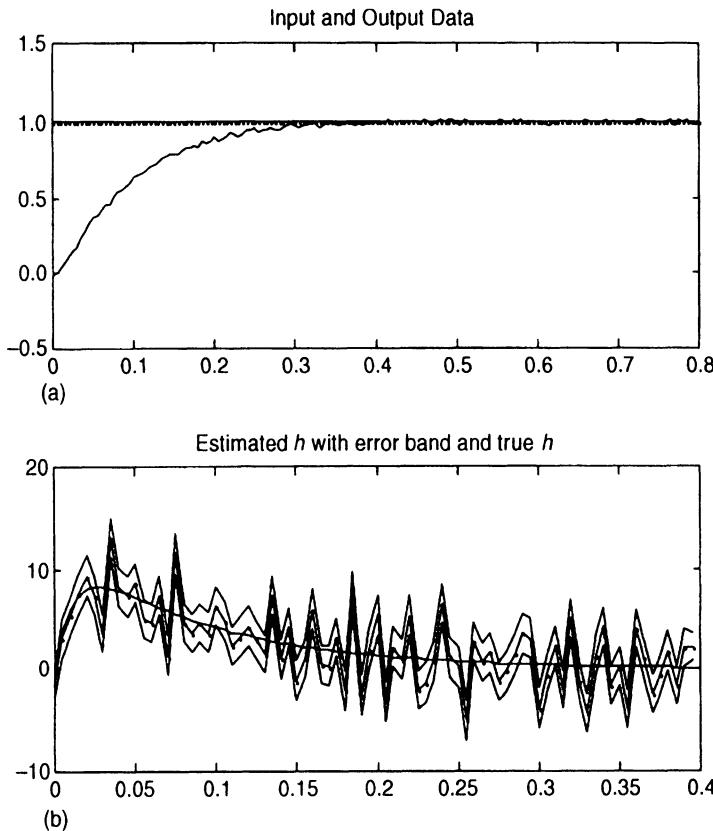
```

>> % Construct observation matrix UU
>> UU = zeros(N,p);
>> for i=1:p,
>> if i==1
>>   UU(:,1)=u;
>> else
>>   UU(:,i) = [zeros(i-1,1)' u(1:N-i+1)']';
>> end
>> end;
>> UU = T*UU;
>>
>> % Construct autocorrelation matrix
>> AA = UU'*UU;
>> b = UU'*y;
>>
>> % Compute estimate of h
>> h = AA\b;
>>
>> % Compute estimated standard errors of h, hse
>> e = y - U*h;
>> sigma = std(e);
>> AAinv = inv(AA);
>> hse = zeros(size(h));
>> for i=1:p,
>> hse(i) = sqrt(AAinv(i,i))*sigma ;
>> end;

```

The above code assumes that the input and output data are contained in the  $N$ -element column vectors  $u$  and  $y$ , respectively. As an illustration of how one can apply the estimation algorithm, we use sample input and output “data” generated by the linear lung mechanics model discussed in Chapters 4 and 5. The following parameter values are assumed:  $L = 0.01 \text{ cm H}_2\text{O s}^2 \text{ L}^{-1}$ ,  $R = 1 \text{ cm H}_2\text{O s L}^{-1}$  and  $C = 0.1 \text{ L cm H}_2\text{O}^{-1}$ . The input in this case is a unit step in  $P_{ao}$ , beginning at time  $t = 0$ . The output is the model response in  $P_A$ . Gaussian white noise is added to the output to simulate the effects of measurement noise. The simulated input and output measurements are displayed in Figure 7.2a; these time series are also contained in the file labeled “*data\_11m.mat*.” The estimated impulse response is shown in Figure 7.2b, along with upper and lower bounds that reflect the estimates plus and minus one standard error. Superimposed on the estimates is the “true” impulse response, which appears as the smooth curve. The fluctuations in the estimated impulse response illustrates how sensitive it is to measurement noise, since in this case, the  $p/N$  ratio of  $\sim \frac{1}{2}$  was large.

Although we have confined the application of this identification method to linear systems here, it is important to point out that Equation (7.3) can be readily extended to take the form of a *Volterra* series, which also contains *nonlinear* dependences of  $y(nT)$  on  $u(nT)$ . This formulation allows us to estimate the parameters that characterize the dynamics of a certain class of open-loop nonlinear systems. However, this topic falls beyond the scope of our present discussion. The interested reader can find further information about this type of



**Figure 7.2** (a) Step change in  $P_{ao}$  and resulting response in  $P_A$  (with noise added). (b) Estimated impulse response with error bounds superimposed on “true” impulse response (smooth curve). Horizontal axes represent time in seconds.

nonlinear system identification in Marmarelis and Marmarelis (1978), Schetzen (1980), and Korenberg and Hunter (1990).

### 7.2.3 Estimation Using Correlation Functions

Starting with Equation (7.3), for any  $m \geq 0$ , if we multiply both sides of the equation by  $u(nT - mT)$ , sum up all  $N - m$  nonzero terms, and then divide through by  $N - m$ , we will obtain

$$\hat{R}_{uy}(mT) = \sum_{k=0}^{p-1} h(kT) \cdot \hat{R}_{uu}(mT - kT) \cdot T + \hat{R}_{ue}(mT) \quad (7.15a)$$

where

$$\hat{R}_{uy}(mT) = \frac{1}{N-m} \sum_{n=m}^{N-1} u(nT - mT)y(nT) \quad (7.16)$$

$$\hat{R}_{uu}(mT - kT) = \frac{1}{N-m} \sum_{n=m}^{N-1} u(nT - mT)u(nT - kT) \quad (7.17)$$

and

$$\hat{R}_{ue}(mT) = \frac{1}{N-m} \sum_{n=m}^{N-1} u(nT - mT)e(nT) \quad (7.18)$$

It should be noted that  $\hat{R}_{uy}$ ,  $\hat{R}_{uu}$ , and  $\hat{R}_{ue}$  in Equations (7.16) through (7.18) represent estimates of the cross-correlation between  $u$  and  $y$ , the autocorrelation in  $u$ , and the cross-correlation between  $u$  and  $e$ , respectively (see Section 5.3.2). We select that solution of  $h$  such that  $\hat{R}_{ue}(mT)$  becomes zero for all values of  $m$ :

$$\hat{R}_{uy}(mT) = \sum_{k=0}^{p-1} h(kT) \cdot \hat{R}_{uu}(mT - kT)T, \quad m = 0, 1, \dots, p-1 \quad (7.15b)$$

Equation (7.15b) may be considered the discrete-time version of Equation (5.27b), and may be solved by applying a little matrix algebra, as in Section 7.2.1.2. We define the following vector and matrix quantities:

$$\hat{\mathbf{R}}_{uy} = [\hat{R}_{uy}(0) \ \hat{R}_{uy}(T) \ \dots \ \hat{R}_{uy}((p-1)T)]' \quad (7.19)$$

and

$$\hat{\mathbf{R}}_{uu} = \begin{bmatrix} \hat{R}_{uu}(0) & \hat{R}_{uu}(T) & \dots & \hat{R}_{uu}((p-1)T) \\ \hat{R}_{uu}(T) & \hat{R}_{uu}(0) & \dots & \hat{R}_{uu}((p-2)T) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{R}_{uu}((p-1)T) & \hat{R}_{uu}((p-2)T) & \dots & \hat{R}_{uu}(0) \end{bmatrix} \quad (7.20)$$

Then, Equation (7.15b) becomes

$$\hat{R}_{uy} = T\hat{\mathbf{R}}_{uu}\underline{h} \quad (7.21)$$

where  $\underline{h}$  is defined by Equation (7.5). Since all elements of  $\hat{\mathbf{R}}_{uy}$  and  $\hat{\mathbf{R}}_{uu}$  can be computed from the input and output data by applying Equations (7.16) and (7.17), the unknown impulse response function is determined through the solution of Equation (7.21) by inverting  $\hat{\mathbf{R}}_{uu}$ :

$$\underline{h} = \frac{1}{T} \hat{\mathbf{R}}_{uu}^{-1} \hat{R}_{uy} \quad (7.22)$$

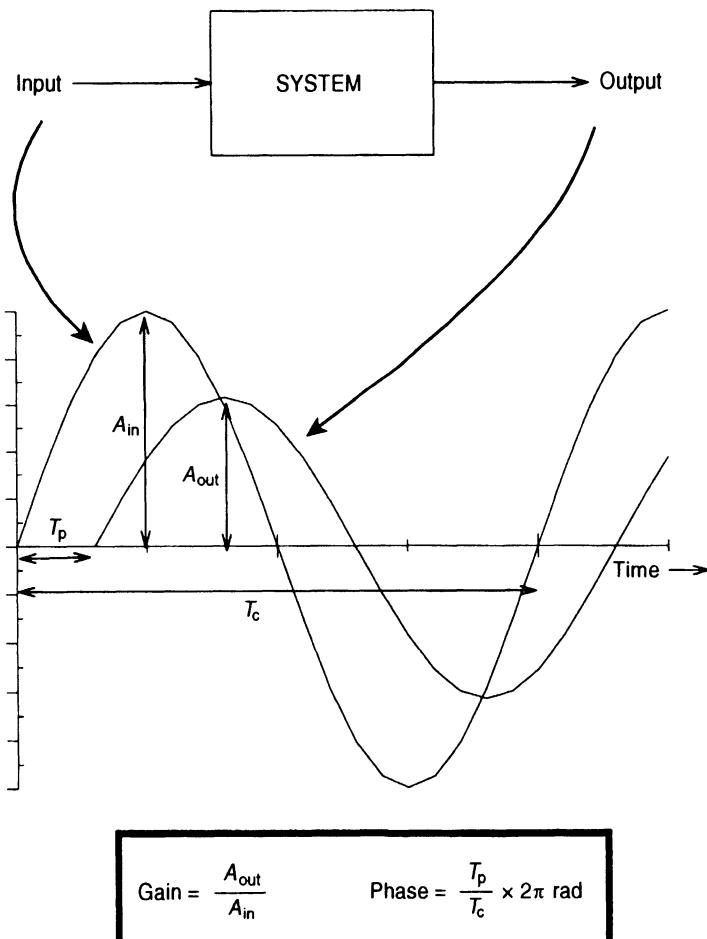
It can be shown that, aside from possible differences in the details of computing the autocorrelation and cross-correlation functions, Equation (7.22) is essentially equivalent to Equation (7.12). As before, the feasibility of applying this approach depends on the invertibility of the autocorrelation matrix  $\hat{\mathbf{R}}_{uu}$ .

#### 7.2.4 Estimation in the Frequency Domain

Since the Laplace transform of the impulse response is the system transfer function, carrying out the system identification process in the frequency domain should, in principle, yield the same results as any of the time-domain methods discussed earlier. The problem of transfer function identification is actually the same as that of estimating the frequency response. The underlying idea is very simple and is illustrated in Figure 7.3. At each frequency in the range of interest, apply a sinusoidal input of known amplitude and phase to the system under study; then measure the resulting output. If the system is linear, the

measured output also will be a sinusoid of the same frequency. The ratio between the magnitude of the output and the magnitude of the input ( $= A_{\text{out}}/A_{\text{in}}$  in Figure 7.3) would yield the system *gain* at that frequency. The phase difference between the output and input waveforms ( $= 2\pi T_p/T_c$ ) would be the system *phase* at that frequency. By repeating this measurement over all frequencies of interest, one would be able to arrive at the frequency response of the system and therefore obtain an estimate of the transfer function.

Although the above method can provide very good estimates of the system transfer function at the frequencies investigated, one major drawback is that the entire identification procedure can be extremely time-consuming and therefore impractical for application in human or animal studies. An alternative would be to employ the spectral analysis technique presented in Section 5.3.2; the basic idea here is that the frequency response is estimated from



**Figure 7.3** Illustration of transfer function identification using sinusoidal inputs. System gain and phase at given frequency  $\omega = 2\pi/T_c$  are as defined above.

the ratio between the input–output cross-spectrum ( $S_{uy}$ ) and the input power spectrum ( $S_{uu}$ ), i.e.,

$$H(\omega_k) = \frac{S_{uy}(\omega_k)}{S_{uu}(\omega_k)} \quad (7.23)$$

where  $\omega_k = 2\pi k/pT$  ( $k = 0, 1, \dots, p - 1$ ), and the spectral quantities are defined in the following way:

$$S_{uy}(\omega_k) = \sum_{m=0}^{p-1} \hat{R}_{uy}(mT) e^{-j\omega_k mT}, \quad k = 0, 1, \dots, p - 1 \quad (7.24)$$

$$S_{uu}(\omega_k) = \sum_{m=0}^{p-1} \hat{R}_{uu}(mT) e^{-j\omega_k mT} \quad k = 0, 1, \dots, p - 1 \quad (7.25)$$

As shown in Equations (7.24) and (7.25),  $S_{uy}(\omega_k)$  and  $S_{uu}(\omega_k)$  are computed by applying the discrete Fourier transform to  $\hat{R}_{uy}$  and  $\hat{R}_{uu}$ , respectively. Equations (7.23) through (7.25) are the discrete-frequency equivalents of Equations (5.29) and (5.30), which were applied to correlation quantities based on continuous time. As mentioned in Section 5.3.2, all values of  $S_{uu}$  must be positive in order for meaningful estimates of the transfer function to be obtained. Whether this condition is attained depends on the form of the input sequence, as we will see in Section 7.3.

Once  $H(\omega)$  has been estimated, it is possible to interpret the results in the context of physiologically meaningful entities if a parametric model is available. As an example of how this can be done, consider the linear lung mechanics model that was discussed in Chapters 4 and 5. From Equation (5.6), the frequency response predicted from this model takes the form

$$H_{\text{model}}(\omega) = \frac{1}{(1 - LC\omega^2) + jRC\omega} \quad (7.26)$$

The values of the lung mechanical parameters ( $R$ ,  $L$ , and  $C$ ) that most closely correspond to the measured frequency response  $H_{\text{meas}}(\omega)$  can be estimated by first defining a “criterion function,”  $J$ , which represents the “distance” between  $H_{\text{model}}$  and  $H_{\text{meas}}$ , and, secondly, by searching for the parameter values that minimize this distance. Since  $H_{\text{model}}$  and  $H_{\text{meas}}$  are complex-valued functions of  $\omega$ , a suitable criterion function might be

$$J = \sum_{k=0}^{p-1} (\text{Re}[H_{\text{meas}}(\omega_k)] - \text{Re}[H_{\text{model}}(\omega_k)])^2 + (\text{Im}[H_{\text{meas}}(\omega_k)] - \text{Im}[H_{\text{model}}(\omega_k)])^2 \quad (7.27)$$

The above expression assumes that frequency response measurements are available at the frequencies  $\omega_k$ , where  $k = 0, 1, \dots, p - 1$ . The methodology for minimizing  $J$  is described in the next section.

### 7.2.5 Optimization Techniques

As we mentioned at the beginning of this chapter, the identification of “gray-box” or “parametric” models consists of two stages. First, the model structure has to be developed, consistent with prior knowledge about the physiological system in question. Frequently, this takes the form of a set of differential equations. Once the model has been formulated, the next task is to estimate the unknown model parameters by minimizing (or maximizing) some criterion that reflects the goodness of fit between the model predictions and the observed output measurements. When dealing with models represented by differential equations of

high order, we mentioned in Section 2.8 that it is generally better, from the viewpoint of numerical stability, to employ a *state-space* framework. Another advantage of employing a state-space model is that the analysis can readily be extended to *nonlinear* systems. We will now illustrate how this system identification technique works by considering our favorite example of the linear lung mechanics model. A more advanced example, involving the analysis of a nonlinear model, is given in Section 7.5.1.

**7.2.5.1 State-space Model Formulation.** The differential equation characterizing the lung mechanics model was derived in Section 4.1 and is given by

$$LC \frac{d^2P_A}{dt^2} + RC \frac{dP_A}{dt} + P_A = P_{ao} \quad (7.28)$$

Since  $P_{ao}$  is the input and  $P_A$  is the output of this system, we make the new variable assignments:

$$y_1 = P_A \quad \text{and} \quad u = P_{ao} \quad (7.29a,b)$$

Also, assume:

$$y_2 = \frac{dP_A}{dt} = \frac{dy_1}{dt} \quad (7.30)$$

Then, we can rewrite Equation (7.28) as:

$$LC \frac{dy_2}{dt} + RCy_2 + y_1 = u \quad (7.31)$$

Using Equations (7.30) and (7.31), rearranging terms, and writing the two equations in matrix form, we obtain:

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{LC} \end{bmatrix} u \quad (7.32a)$$

If we define:

$$\underline{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (7.33)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \quad (7.34)$$

and

$$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{LC} \end{bmatrix} \quad (7.35)$$

Equation (7.32a) becomes:

$$\frac{d}{dt} \underline{\mathbf{y}} = \mathbf{A}\underline{\mathbf{y}} + \mathbf{B}u \quad (7.32b)$$

Thus, we have converted the second-order scalar differential equation (Equation (7.31)) into the equivalent first-order matrix state equation. This type of equation can be conveniently

solved by numerical integration using one of the MATLAB ordinary differential equation solver functions: “ode45,” “ode23,” “ode113,” “ode15s,” and “ode23s.” An even easier way is to construct a state-space representation of the model within MATLAB using the “ss” function, and then use the “lsim” function (see Section 4.6) to generate the model response to a given input waveform  $u(t)$ . The MATLAB script file (provided as the file “sss\_llm.m”) that performs these tasks is displayed below.

```
>> A = [0 1; -1/L/C -R/L];
>> B = [0 1/L/C]';
>> t = [0:0.005:0.8]';
>> u = ones(size(t));

% Construct the system using state-space formulation
>> Hs = ss(A,B,[1 0],0);

% Solve state space equation using lsim and plot results
>> y = lsim(Hs,u,t);
>> plot(t,u,t,y)
```

In the above MATLAB script file, note that we have assumed the following companion “observation equation”:

$$\hat{y} = \mathbf{D}\mathbf{y} + \mathbf{E}u \quad (7.36)$$

where  $\hat{y}$  represents the measured output of this system, i.e.,  $P_A$ . In this case, we are able to measure the state variable  $y_1 (= P_A)$  directly. Thus, here, we have:

$$\mathbf{D} = [1 \ 0] \quad \text{and} \quad \mathbf{E} = 0 \quad (7.37a,b)$$

This accounts for the last two items in the argument list of the function “ss” in the above MATLAB script. One other detail is that the matrices  $\mathbf{A}$  and  $\mathbf{B}$  do not need to be evaluated directly. If the transfer function of the model is available, then the MATLAB function “tf2ss” can be used to convert the system representation from transfer function format to state-space format:

```
>> [A, B, D, E] = tf2ss(num, den);
```

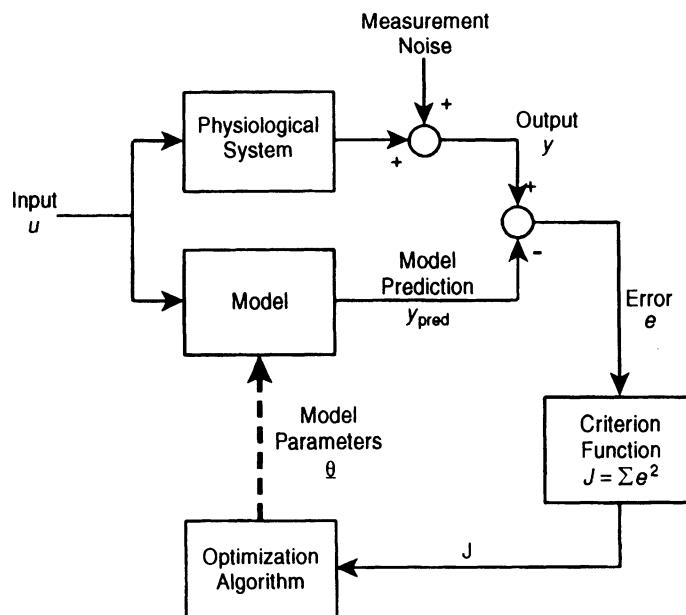
**7.2.5.2 Optimization algorithm.** Having constructed the model, the next step is to select the means by which the response of the model to a given input sequence can be compared to the response of the physiological system to the same input. The comparison is made through the use of a *criterion function* that provides a measure of the *goodness of fit* between the two time series. There are many possible candidates for the criterion function, but the one most commonly employed is the sum of squares of the differences between the measured and predicted outputs:

$$J = \sum_{n=0}^{N-1} \{y(nT) - y_{\text{pred}}(nT)\}^2 = \sum_{n=0}^{N-1} e(nT)^2 \quad (7.38)$$

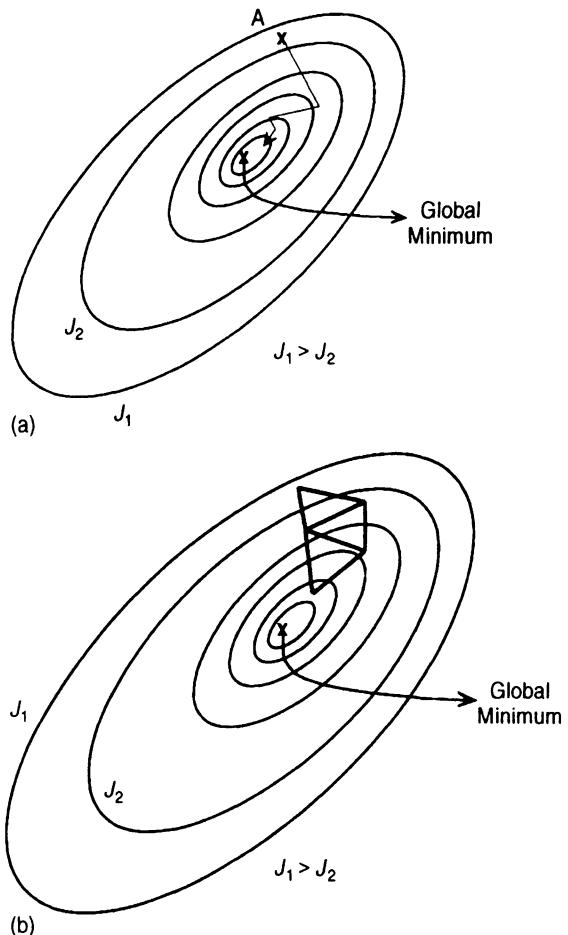
This is the same expression as that presented in Equation (7.9) in Section 7.2.2.

With the criterion function having been defined, the problem of parameter estimation becomes transformed into a problem of optimization, where the objective is to find the combination of parameter values that minimizes the criterion function. The entire scheme of parameter estimation is illustrated in the schematic block diagram shown in Figure 7.4. It should be noted that if the model selected provides an accurate representation of the dynamics of the real system, then the residual errors  $\{e(nT), n = 0, 1, \dots, N - 1\}$  should closely reflect the measurement noise affecting the output. On the other hand, if the selected model is largely “wrong” and does not provide an adequate description of the output dynamics, there will be a significant contribution from *structural errors* as well. This is one of the major drawbacks of opting to employ a “structural” model: erroneous information about the dynamics of the underlying physiology can translate into large errors in the parameter estimates and/or an inability of the model predictions to “fit” the data well.

The choice of the algorithm employed to perform the minimization of the criterion function is also important. There is a large array of algorithms available, but it is not within the scope of this chapter to examine all or even most of these. The most commonly applied methods employ the *gradient descent* approach. These methods are best explained by considering a problem in which two parameters need to be estimated. We will refer to these two parameters as  $\theta_1$  and  $\theta_2$ . Since  $J$  is a function of  $\theta_1$  and  $\theta_2$ , evaluating  $J$  over selected ranges of  $\theta_1$  and  $\theta_2$  would yield a surface in a 3-dimensional space with the Cartesian axes formed by  $\theta_1$ ,  $\theta_2$ , and  $J$ . Suppose the surface looks like the contour map shown in Figure 7.5a, where each contour corresponds to a uniform value of  $J$ . In the gradient descent approach, we start off with an initial guess of the parameters, represented as the point  $A$ . Then, information about the slope of the local terrain is obtained, and based on this information, we move down the slope along the direction of “steepest descent.” The size of the step taken in this direction differs with the different gradient descent methods, with some methods using information about the curvature (i.e., second derivatives) of the surface as well.



**Figure 7.4** Schematic diagram of the optimization approach to parameter estimation.



**Figure 7.5** Methods for finding the minimum of the criterion function surface: (a) steepest gradient method; (b) simplex method.

One drawback of the gradient approach is the need to evaluate the derivatives of  $J$  with respect to all the parameters being estimated at every iteration step. Furthermore, these methods are generally quite susceptible to producing solutions that correspond to false local minima, if the  $J$ -surface is highly irregular, as would be the case when the signal-to-noise ratio is low. A popular alternative, which does not require any derivative computations at all, is the *Nelder–Mead simplex* algorithm. For a 3-parameter problem, the *simplex* takes the form of a tetrahedron, while for the 2-parameter problem, it is a triangle. Figure 7.5b shows the same criterion function surface discussed above, together with the simplex (triangle) and how the shape and position of the triangle moves over the course of a few iterations. The vertices of the triangle represent the three points on the  $J$ -surface that are known at any given iteration of the algorithm. Starting at the initial three points, the triangle is reflected over the two vertices with the lowest  $J$ -values and the height of the triangle is expanded or contracted so that the remaining vertex is located at the point of lowest  $J$ -value. Next, the triangle is reflected over the two of the three vertices that have the lowest  $J$ -values, and the new third vertex is found by stretching or shrinking the height of the triangle. This process is repeated until some tolerance for convergence towards the minimum is met.

The MATLAB function “`fmins`” employs the Nelder–Mead simplex algorithm to determine the minimum point of a given multidimensional function. The MATLAB script

(contained in the m-file named “*popt\_llm.m*”) presented below shows an example of how “*fmins*” can be used to estimate the values of the two unknown parameters in the state-space formulation of the linear lung mechanics model (Section 7.2.5.1).

```
>> global u y
>> theta_init(1)= input(' Enter initial value of 1st parameter >>');
>> theta_init(2)= input(' Enter initial value of 2nd parameter >>');

% Perform optimization to minimize the objective function J
% defined by the function "fn_llm"
>> [theta,options] = fmins('fn_llm',theta_init);
```

Two items are required as inputs to “*fmins*”. The first is a user-defined function that defines the model being employed and returns to “*fmins*” the value of the criterion function at each iteration in the optimization process. In our particular example, we have named this function “*fn\_llm*”. The second input is a column vector (“*theta\_init*”) containing an initial guess of the parameters to be estimated. “*fmins*” produces two sets of outputs. The first set contains the estimated parameter vector (“*theta*”). The second set of outputs (“*options*”) contains information about the minimization process. For instance, the 10th element of “*options*” contains the total number of iterations or function calls in the optimization run. As well, in this example, the data file “*data\_llm.mat*” has to be loaded prior to running “*popt\_llm*” so that the input and output data is present in the workspace as vectors *u* and *y*, respectively, for “*fmins*” to work on. Since the function “*fn\_llm*” must also use these data, the “*global*” declaration is included in both “*popt\_llm*” and “*fn\_llm*” to make *u* and *y* universally accessible. The relevant portion of the MATLAB code for “*fn\_llm*” is given below, with the complete listing given in the m-file “*fn\_llm.m*”.

```
>> function J = fn_llm(theta)
>> global u y

>> A = [0 1; -theta(1) -theta(2)];
>> B = [0 theta(1)]';
>> Hs = ss(A,B,[1 0],0);
>> ypred = lsim(Hs,u,t);
>> e = y - ypred;
>> J = sum(e.^2);
```

It should be noted from the script for “*fn\_llm*” that the two parameters being estimated,  $\theta_1$  and  $\theta_2$ , correspond to the lung mechanical parameters  $1/LC$  and  $R/L$ , respectively (see Equations (7.34) and (7.35)). Application of this algorithm to the simulated data given in “*data\_llm.mat*” leads to the parameter estimation results shown below. The estimated parameter values of 1006.3 and 100.8 agree closely with the “true” values for  $1/LC$  and  $R/L$  of 1000 and 100, respectively.

```
Final Parameter Values :  
1.0e+003 *  
1.0063  
0.1008  
Total Number of Iterations:  
198
```

## 7.3 PROBLEMS IN PARAMETER ESTIMATION: IDENTIFIABILITY AND INPUT DESIGN

### 7.3.1 Structural Identifiability

The problem of structural identifiability is intimately coupled to the problem of model-building. In theory, if knowledge about the underlying physiology of the system in question is available, it should be possible for us to translate this knowledge into a parametric model by applying the basic laws of physics and chemistry. The more we know about the system, the more details we will be able to add to the model. In general, a more detailed and complex model would be expected to account for a greater range of observations under a larger variety of conditions. However, the price that one has to pay for the increased model complexity is the emergence of more model *parameters*, the values of which have to be assumed or estimated. In the several models that we have discussed in previous chapters, we assumed the parameter values to be known. For example, in the linear lung mechanics model, we assumed values for  $R$ ,  $L$ , and  $C$  that were considered “representative” of the population of subjects with normal lungs. This assumption ignores the fact that there is a considerable degree of variability in these lung mechanical parameters across subjects that one can consider “normal.” On the other hand, we could choose to estimate the parameters in each individual subject. The problem of structural identifiability arises when the information that is required for the parameter estimation process is incomplete. This could be due to the inaccessibility of certain signals or the lack of dynamic content in the stimulus.

As an example, consider the differential equation (Equation (7.28)) that characterizes the linear lung mechanics model. Here, there are three unknown parameters:  $R$ ,  $L$ , and  $C$ . However, the mathematical structure of this model turns out to be such that the parameters only appear as paired combinations of one another:  $LC$  and  $RC$ . As a consequence, the dynamics of the model are determined by only two parameters ( $LC$  and  $RC$ ) and not by the original three ( $R$ ,  $L$ , and  $C$ ). This fact again becomes evident when one looks at the state-space formulation of the model in Equation (7.32a). Here, only two independent parameters determine the solution (i.e., dynamics) for the vector  $y$ , and these are  $1/LC$  and  $R/L$ . Thus, it is clear that, using only measurements of  $P_A$  and  $P_{ao}$ , the linear lung mechanics model is not completely identifiable in terms of all three parameters— $R$ ,  $L$ , and  $C$ . We should stress that this assertion on identifiability (or rather, the lack of it) holds true regardless of whether noise is present or absent in the measurements. On the other hand, this model could become fully identifiable if an additional channel of measurement, such as airflow, were to become available. For instance, one could estimate  $C$  separately from static changes, e.g., from the ratio of the change in lung volume resulting from an applied change in pressure. Then, by combining this additional piece of information with the two parameters that can be estimated from the step-response in  $P_A$ , we would be able to identify all three original model parameters.

### 7.3.2 Sensitivity Analysis

A model that has been found to be structurally identifiable may still turn out to be unidentifiable in practice, if the parameter estimation process is sufficiently degraded by the presence of measurement noise. Therefore, having arrived at a structurally identifiable model, the next test that we should subject the model to is the determination of whether the parameters that need to be estimated are resolvable in the presence of noise. Since the parameter estimation process requires us to find the lowest point on the multidimensional surface of the criterion function, it follows that parameter identifiability depends heavily on the quality of the  $J$ -surface. Figure 7.6 illustrates this statement with the help of two hypothetical examples that assume the case involving only a single parameter ( $\theta_1$ ). In both cases, we also assume that the presence of measurement noise limits the resolvability of changes in  $J$  to a value  $\Delta J$ . In Figure 7.6a, there is a deep minimum. The error in the parameter estimate ( $\Delta\theta_1$ ) made in arriving at a solution that is located at a criterion function value  $\Delta J$  above the global minimum is small. On the other hand, in Figure 7.6b, the  $J$ -surface contains a very shallow minimum. In this case, the effect of the same amount of measurement noise is a much larger error in the parameter estimate. What distinguishes case (a) from case (b) in this example is the fact that in case (a), a given change in the parameter value leads to a large (and therefore, highly observable) change in the model output or  $J$ . Thus, in case (a), the model possesses *high sensitivity* to parameter variations, whereas in case (b) sensitivity is low.

The inverse relationship between sensitivity and parameter estimation error can be demonstrated analytically. We begin by recalling the definition of the criterion function (see Equation (7.38)) but rewriting it in vector form:

$$J = (\underline{y} - \underline{y}_{\text{pred}})'(\underline{y} - \underline{y}_{\text{pred}}) \quad (7.39)$$

Differentiating  $J$  with respect to the parameter vector  $\underline{\theta}$ , we obtain

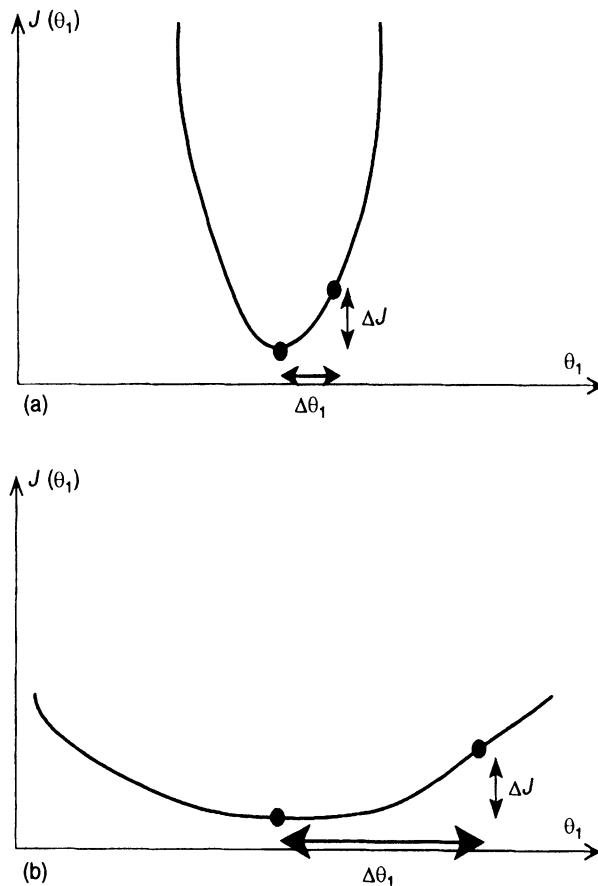
$$\frac{\partial J}{\partial \underline{\theta}} = -(\underline{y} - \underline{y}_{\text{pred}})' \frac{\partial \underline{y}_{\text{pred}}}{\partial \underline{\theta}} \quad (7.40)$$

Note that the derivative on the right-hand side of Equation (7.40) is a  $N \times p$  matrix, the elements of which represents the effect of a small change in each parameter on the model output. Thus, we can refer to this entity as the *sensitivity matrix*,  $S$ , i.e.:

$$S = \frac{\partial \underline{y}_{\text{pred}}}{\partial \underline{\theta}} \quad (7.41)$$

Suppose  $\underline{\theta}^*$  represents the parameter vector at the global minimum point on the  $J$ -surface. Then, by applying a Taylor series expansion and keeping *only first-order terms*, the model output in the vicinity of the minimum point can be expressed as

$$\underline{y}_{\text{pred}}(\underline{\theta}) = \underline{y}_{\text{pred}}(\underline{\theta}^*) + \frac{\partial \underline{y}_{\text{pred}}}{\partial \underline{\theta}} \Big|_{\underline{\theta}^*} (\underline{\theta} - \underline{\theta}^*) = \underline{y}_{\text{pred}}(\underline{\theta}^*) + S_{\underline{\theta}^*}(\underline{\theta} - \underline{\theta}^*) \quad (7.42)$$



**Figure 7.6** Relationship between sensitivity to parameter variations and parameter estimation error: (a) high sensitivity; (b) low sensitivity.

where  $\mathbf{S}_{\underline{\theta}^*}$  denotes the matrix  $\mathbf{S}$  evaluated at the minimum point. At the minimum point,  $\partial J / \partial \underline{\theta}$  in Equation (7.40) becomes a null vector. Thus, substituting Equation (7.42) into Equation (7.40), we obtain

$$-\frac{\partial J}{\partial \underline{\theta}} = 0 = \{\mathbf{y} - \mathbf{y}_{\text{pred}}(\underline{\theta}^*)\}' \mathbf{S}_{\underline{\theta}^*} - (\underline{\theta} - \underline{\theta}^*)' \mathbf{S}_{\underline{\theta}^*}' \mathbf{S}_{\underline{\theta}^*} \quad (7.43)$$

which can be rearranged to yield the following expression for the parameter estimate error:

$$(\underline{\theta} - \underline{\theta}^*)' = \mathbf{e}^* \mathbf{S}_{\underline{\theta}^*} (\mathbf{S}_{\underline{\theta}^*}' \mathbf{S}_{\underline{\theta}^*})^{-1} \quad (7.44)$$

where  $\mathbf{e}^*$  represents the vector that contains the residual errors between the measurements  $\mathbf{y}$  and the predicted output values  $\mathbf{y}_{\text{pred}}(\underline{\theta}^*)$ . Left-multiplying both sides of Equation (7.44) with  $(\underline{\theta} - \underline{\theta}^*)$  and applying the expectation operator (see Equation (5.28)) to both sides of the resulting equation, we get

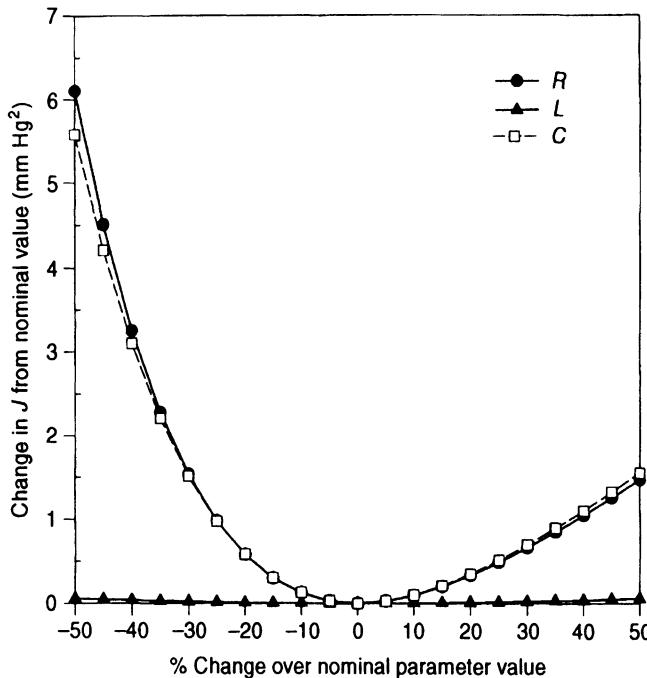
$$\mathbf{P}_{\underline{\theta}} \equiv E[(\underline{\theta} - \underline{\theta}^*)(\underline{\theta} - \underline{\theta}^*)'] = (\mathbf{S}_{\underline{\theta}^*}' \mathbf{S}_{\underline{\theta}^*})^{-1} \mathbf{S}_{\underline{\theta}^*}' E[\mathbf{e}^* \mathbf{e}^{*'}] \mathbf{S}_{\underline{\theta}^*} (\mathbf{S}_{\underline{\theta}^*}' \mathbf{S}_{\underline{\theta}^*})^{-1} \quad (7.45)$$

where  $\mathbf{P}_{\underline{\theta}}$  is also known as the *parameter error covariance matrix*. The diagonal elements of  $\mathbf{P}_{\underline{\theta}}$  contain the variances of all  $p$  parameters in  $\underline{\theta}$ , whereas the off-diagonal elements represent the cross-covariances between the different paired combinations of the parameters. If we assume the sequence of residual errors to be white, i.e., the present error is uncorrelated with past or future errors, then the matrix  $E[\mathbf{e}^* \mathbf{e}^{*'}]$  reduces to the identity matrix scaled by a factor equal to the variance,  $\sigma^2$ , of the residual errors. Thus, Equation (7.45) simplifies to

$$\mathbf{P}_{\underline{\theta}} = \sigma^2 (\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*})^{-1} \quad (7.46)$$

From Equation (7.46), it is important to note that each element of the symmetric matrix  $\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*}$  reflects the change in model output resulting from small changes in all possible pairings of the parameters. If changes in one or more of the parameters have no effect on the model output (zero sensitivity), then one or more columns and rows of  $\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*}$  will be zero; as a result,  $\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*}$  will be singular and the parameter errors will be infinite. This occurs when the model is *not structurally identifiable*. In structurally identifiable models,  $\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*}$  can still become close to singular if there are strong interdependences between some of the parameters; in this case, there will be strong correlations between columns or rows of matrix  $\mathbf{S}'_{\underline{\theta}^*} \mathbf{S}_{\underline{\theta}^*}$ . Inversion of this close-to-singular matrix will yield variance and covariance values in  $\mathbf{P}_{\underline{\theta}}$  that are unacceptably large. However, it is important to bear in mind from Equation (7.46) that, even under circumstances where model sensitivity is high, it is still possible for the parameters to be poorly estimated if the variance of the measurement noise ( $\sigma^2$ ) is very large.

Equation (7.46) provides lower-bound estimates of the variances and cross-covariances associated with the model parameters when these are estimated from noisy measurements. However, computation of the  $\mathbf{P}_{\underline{\theta}}$  matrix is based on local changes in the vicinity of the optimal set of parameter values. A common alternative method of assessing model sensitivity is to base the calculations over a larger range of parameter value changes. In this approach, the criterion function  $J$  is evaluated over a selected span of values (say,  $\pm 50\%$ ) for each parameter in turn, while holding the rest at their nominal values. Ideally, the “nominal” or “reference” values selected should correspond to the optimal set  $\underline{\theta}^*$ . The form of the criterion function  $J$  is the same as that given in Equation (7.39), except that, in this case, the vector of observations  $\mathbf{y}$  is replaced by  $\mathbf{y}_{\text{ref}}$ , where the latter represents the model predictions when the parameters are at their nominal values. Here,  $\mathbf{y}_{\text{pred}}$  corresponds to the vector of model predictions at any of the parameter combinations ( $\neq \underline{\theta}^*$ ) being evaluated. An example of this type of sensitivity analysis is shown in Figure 7.7 for the linear lung mechanics model. The nominal parameter set in this case is:  $R = 1 \text{ cm H}_2\text{O s L}^{-1}$ ,  $L = 0.01 \text{ cm H}_2\text{O s L}^{-2}$  and  $C = 0.1 \text{ L cm H}_2\text{O}^{-1}$ . The model is assumed to be perturbed by a unit step in  $P_{\text{ao}}$ . The plot for  $R$ , for instance, shows changes in  $J$  that would result if  $R$  were to be varied over the range  $0.5\text{--}1.5 \text{ cm H}_2\text{O s L}^{-1}$ , while  $L$  and  $C$  are kept at their nominal values. The model output is reasonably sensitive to changes in  $R$  and  $C$ , but virtually insensitive to changes in  $L$ . This kind of “flatness” in the sensitivity curve provides a good indication that at least one of the parameters will be not identifiable. This conclusion is consistent with our analysis of structural identifiability of this model in Section 7.3.1. The sensitivity results in Figure 7.7 were generated by the MATLAB script file “*sensan1.m*” (which also calls the function “*fn\_rlc.m*”).



**Figure 7.7** Sensitivity of the linear lung mechanics model to variations in the model parameters about their nominal values ( $R = 1$ ,  $L = 0.01$ ,  $C = 0.1$ ). Flat curve for  $L$  (i.e. very low sensitivity) suggests identifiability problems for this parameter.

### 7.3.3 Input Design

The result represented by Equation (7.46) is valid for any general parametric model. Comparison of this result with Equation (7.13) shows a striking similarity between the two equations. This similarity is by no means coincidental. In fact, Equation (7.13) represents a special case of Equation (7.46) when the assumed “model” is simply the impulse response of the system under study. To demonstrate this, note that the vector  $\underline{h}$  containing the sampled impulse response is the unknown parameter vector  $\underline{\theta}$  that we would like to estimate. Thus, the model predictions are given by

$$y(nT)_{\text{pred}} = \sum_{k=0}^{p-1} \theta_k \cdot u(nT - kT) \cdot T, \quad n = 0, 1, \dots, N-1 \quad (7.47)$$

where

$$\theta_k = h(kT) \quad (7.48)$$

The sensitivity of the  $n$ th output value to changes in the  $k$ th parameter is

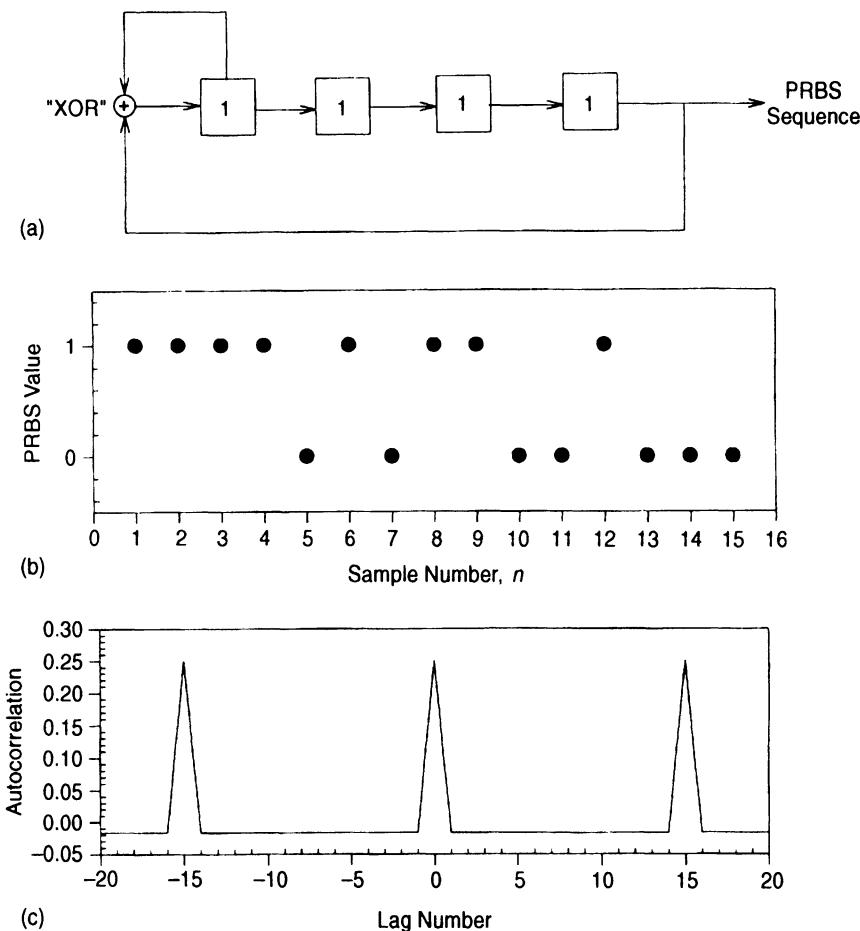
$$S_{nk} \equiv \frac{\partial y(nT)_{\text{pred}}}{\partial \theta_k} = u(nT - kT) \cdot T, \quad n = 0, 1, \dots, N-1; \quad k = 0, 1, \dots, p-1 \quad (7.49)$$

Looking back at Equation (7.7), it can be easily seen that the matrix  $\mathbf{U}$  is simply a special case of the sensitivity matrix  $\mathbf{S}$ . Consequently, we have the following equality:

$$\mathbf{S}'_{\theta^*} \mathbf{S}_{\theta^*} = \mathbf{U}' \mathbf{U} \quad (7.50)$$

While it was not specifically mentioned in the previous section, it is clear from Equation (7.50) that the  $p \times p$  matrix  $\mathbf{S}'_{\theta^*} \mathbf{S}_{\theta^*}$  is a function of the input time-course. This implies that if the researcher has control over the type of stimulus that can be administered to the system in question, it should be possible to design the input waveform in such a way as to best “condition” the matrix  $\mathbf{S}'_{\theta^*} \mathbf{S}_{\theta^*}$  so that the elements of its inverse can be minimized. From linear algebra, we know that a matrix is singular if there is linear dependence between any two or more of its columns (or rows). The best-case scenario for matrix inversion occurs when the matrix to be inverted is diagonal and all the diagonal elements are nonzero. For the matrix  $\mathbf{U}' \mathbf{U}$  to become diagonal, it would be necessary to choose an input time course in which any sample in the waveform is uncorrelated with all other samples. Another way of saying this is that the input waveform should have zero autocorrelation over all lags, except at the zeroth lag (which simply measures how correlated the signal is with itself). One type of input waveform that has this kind of autocorrelation function is white noise. This is one of the reasons for the popularity of white noise as a test input. Another reason relates to the fact that the white noise time series also has a power spectrum that is essentially flat over a broad range of frequencies. This *persistently exciting* kind of stimulation allows the system to be probed over a larger range of dynamic modes.

Although white noise has been employed as a test input in many studies investigating various neural systems, it has not been used as much for identifying other physiological systems. A major reason for this is the practical difficulty of implementing this kind of input forcing. The *pseudo-random binary sequence* (PRBS) offers an attractive alternative that is very easy to implement and can lead to good estimation results in many applications. The PRBS is so named because the time series produced is actually periodic with a cycle duration of  $N + 1$  samples, if  $N$  is the total number of points in the sequence. However, within one period of this series, each sample is virtually uncorrelated with other samples. One of the most commonly used methods for generating the PRBS employs binary shift-registers with feedback. Figure 7.8a displays a 4-stage shift-register. The process begins with all stages assigned a value of 1. Then, at the end of each time-step ( $T$ ), the value contained in each stage is moved to the right by one stage. The value in the rightmost stage of the shift-register (=“1”) becomes the first value of the PRBS. At the same time, this value is fed back toward the first stage and is added (or more precisely, “XOR-ed”) to the value originally in the first stage. In this case, applying Boolean arithmetic, we get:  $1 + 1 = 0$ . Thus, at the end of the first time-step, the values in the shift-register are “0111”. During the next time-step, the “1” value in the rightmost stage is moved to the right and becomes the second value in the PRBS. At the same time, this value is fed back to the first stage and added to its original value:  $1 + 0 = 1$ . The new result is assigned to the first stage. Thus, at the end of the second time-step, the values in the shift-register are “1011”. This process continues on until the values in the shift-register revert to “1111”, which was what it had started with. It can be easily shown that the 4-stage shift register assumes the “1111” value at the end of the 16th time-step and the whole sequence repeats itself. The output of this process is a 15-point sequence with randomlike properties, as depicted in Figure 7.8b. The autocorrelation function of this kind of sequence approximates that of white noise up to a maximum lag number of 14, as is shown in Figure 7.8c. However, beyond this range, it is clear that the sequence is periodic. The maximum autocorrelation value for a PRBS signal of amplitude  $A$  is  $A^2$ , and the minimum



**Figure 7.8** (a) Shift register method for generating a 15-point pseudorandom binary sequence. (b) The 15-point PRBS signal generated from (a). (c) Theoretical autocorrelation function of the 15-point PRBS signal.

value is  $-A^2/N$ . Thus, in our example, the maximum and minimum values turn out to be 0.25 and  $-0.01667$ , respectively. Although the PRBS example shown here is based on a 4-stage shift-register, the latter may be extended to more stages. For an  $m$ -stage shift-register, the total output sequence will consist of  $2^m - 1$  “random” values. The PRBS signal displayed in Figure 7.8b was generated by executing the MATLAB script file “prbs.m”.

Using the PRBS as an input can lead to a dramatic simplification of the correlation method of system identification (see Section 7.2.3). If an  $N$ -sample PRBS input of amplitude  $A$  is employed, the autocorrelation matrix  $\hat{\mathbf{R}}_{uu}$  (of size  $N \times N$ ) becomes

$$\hat{\mathbf{R}}_{uu} = \frac{A^2}{N} \begin{bmatrix} N & -1 & \dots & -1 \\ -1 & N & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & N \end{bmatrix} \quad (7.51)$$

This matrix can easily be inverted, taking the following form:

$$\hat{\mathbf{R}}_{uu}^{-1} = \frac{N}{A^2(N+1)} \begin{bmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 2 \end{bmatrix} \quad (7.52)$$

One can verify that the right-hand side of Equation (7.52) is the inverse of  $\hat{\mathbf{R}}_{uu}$  by multiplying this by the right-hand side of Equation (7.51) and showing that the result yields the identity matrix.

Then, applying Equation (7.22), we can obtain the impulse response vector:

$$\mathbf{h} = \frac{N}{A^2(N+1)T} \begin{bmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 2 \end{bmatrix} \begin{bmatrix} \hat{R}_{uy}(0) \\ \hat{R}_{uy}(T) \\ \vdots \\ \hat{R}_{uy}((N-1)T) \end{bmatrix} \quad (7.53)$$

By evaluating the right-hand side of Equation (7.53), we can decompose the above matrix equation into the following set of equations:

$$h(kT) = \frac{N}{A^2(N+1)T} \left( \hat{R}_{uy}(kT) + \sum_{i=0}^{N-1} \hat{R}_{uy}(iT) \right), \quad k = 0, 1, \dots, N-1 \quad (7.54)$$

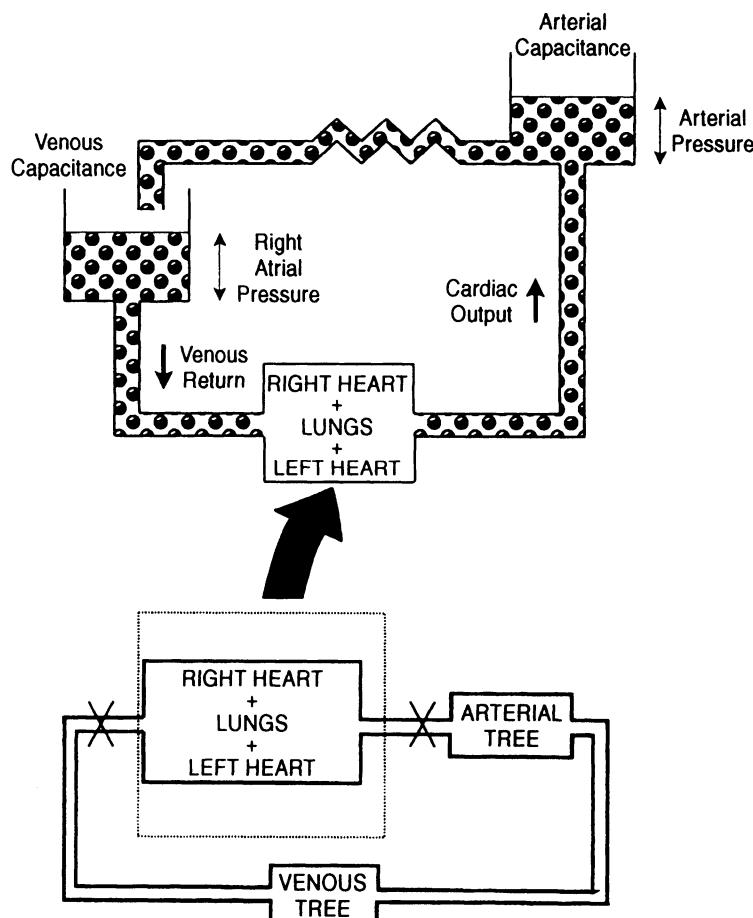
The expression for  $h(kT)$  in Equation (7.54) makes it necessary only to compute the cross-correlation between the input and output sequences. Explicit matrix inversion is thereby averted. However, a serious practical limitation of Equation (7.54) is that the errors associated with the estimates of the impulse response can be unacceptably large if the input and output measurements are very noisy, since  $N$  values of  $h(kT)$  have to be estimated from  $N$  pairs of input-output data (see Section 7.2.2). A good example of the application of this technique to the identification of a physiological system is given in Sohrab and Yamashiro (1980).

## 7.4 IDENTIFICATION OF CLOSED-LOOP SYSTEMS: “OPENING THE LOOP”

The system identification methods discussed in Section 7.3 were based implicitly on the assumption of an open-loop system: the stimulus (input) to the system was assumed to be unaffected by the response (output). However, since most physiological control processes operate under closed-loop conditions, researchers have applied a variety of techniques to “open the loop” by isolating the components of interest from other components that comprise the entire system. In some cases, “opening the loop” has meant literally that: the subsystem of interest was surgically separated from the rest of the system. Denervation, ablation of certain focal areas, and the redirection of blood flow have become standard techniques in physiological investigations. Another group of methods have been less invasive, involving the use of pharmacological agents to minimize or eliminate potentially confounding influences while the component of interest is studied. A third class of techniques apply clever, noninvasive experimental manipulations to the intact physiological system in order to open the loop functionally rather than physically or pharmacologically. In this chapter, we will review several classic examples that represent the wide spectrum of these methods.

### 7.4.1 The Starling Heart–Lung Preparation

In Section 3.5, we discussed a simple closed-loop model of cardiac output regulation, consisting of essentially two major subsystems, one comprising of the heart and pulmonary circulation, and the other representing the systemic circulation. The now legendary experiments by Patterson, Piper, and Starling (1914) provided the first systematic characterization of the former subsystem, thereby enabling the measurement of the intrinsic response of the heart to changes in venous return and arterial blood pressure. As illustrated schematically in Figure 7.9, the heart and lungs were surgically isolated from the rest of the systemic circulation. By connecting the right atrium to a reservoir of blood placed above it and controlling the flow of blood from the reservoir to the heart, the researchers were able to artificially vary the right atrial pressure. Blood ejected from the left ventricle was led to an arterial capacitance and then through an adjustable resistance (“Starling resistor”) back to the venous reservoir after being heated to body temperature. Adjustment of the Starling resistor or the vertical position of the arterial capacitance allowed the researchers to control arterial (or aortic) pressure. In this way,

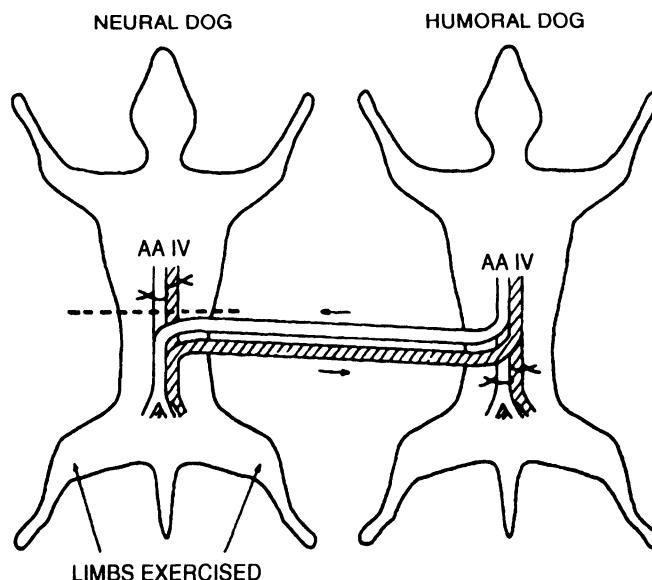


**Figure 7.9** Schematic illustration of the way in which Starling and coworkers “opened the loop” to study the control of cardiac output.

systematic changes in right atrial pressure and arterial pressure were related to the corresponding cardiac output. These data formed the basis of Guyton's cardiac function curves (see Section 3.5.1).

### 7.4.2 Kao's Cross-Circulation Experiments

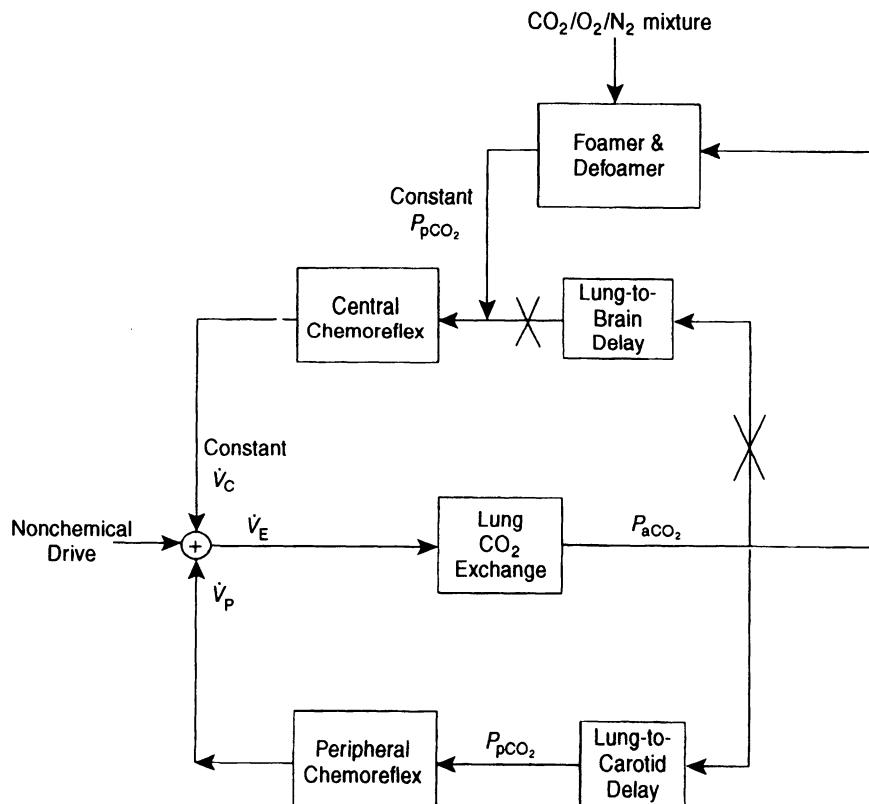
Kao and Ray (1954) performed experiments on anesthetized dogs to determine whether the increase in cardiac output observed during exercise was due to neural or humoral (blood-borne) factors. In order to separate the neural from humoral effects, their experiments were designed in the following way. In each experiment, two anesthetized dogs were used. The hind-limbs of the "neural dog" were stimulated electrically so that muscular work was induced. However, arterial blood perfusing the hind-limbs of this dog came from the second dog, and venous blood leaving the limbs was directed back to the "humoral dog." The basic experimental design is displayed in Figure 7.10. The authors hypothesized that: (1) if the exercise-induced cardiac output increase was due solely to neural feedback from the exercising limbs, the "neural dog" would continue to show this increase while the "humoral dog" should not respond at all; and (2) if the exercise-induced cardiac output increase was due solely to humoral factors, the "humoral dog" should show this increase, while there should be no response in the "neural dog." Based on the results of nine pairs of these animals, it was found that cardiac output increased significantly in both "neural" and "humoral" dogs. This led the authors to conclude that both neural and humoral factors are involved in the regulation of cardiac output during muscular activity.



**Figure 7.10** Kao's experimental design for separating neuromuscular feedback from humoral effects on exercise-induced hyperpnea. Reproduced from Kao and Ray (1954).

### 7.4.3 Artificial Brain Perfusion for Partitioning Central and Peripheral Chemoreflexes

In Section 6.7, we examined the stability properties of a model of the chemoreflex regulation of ventilation. The analysis showed that the gains and time constants associated with the central and peripheral chemoreflexes are important determinants of respiratory stability. The question of being able to measure the dynamics of these two chemoreflexes in isolation from one another was addressed by Berkenbosch and colleagues (1979) in a series of experiments that employed the clever technique of artificial brain perfusion. This method is illustrated schematically in Figure 7.11. In anesthetized cats, the researchers directed blood from one of the femoral arteries through an extracorporeal circuit in which the blood was equilibrated in a foamer with a gas mixture of known composition, defoamed, and then returned to the cat through a cannulated vertebral artery. The other vertebral artery was clamped, so that the brain was perfused only by the blood leaving the extracorporeal circuit. This allowed the  $P_{CO_2}$ ,  $P_{O_2}$ , and pH of the blood perfusing the medullary chemosensitive regions to be maintained at constant levels set by the researchers. This effectively “opened” the central chemoreflex loop. Consequently, the effects of dynamic changes in arterial  $P_{CO_2}$  or  $P_{O_2}$  (produced by inhalation of hypercapnic or hypoxic gas mixtures) on the peripheral chemoreflex contribution to ventilation could be measured in isolation from the central contribution.



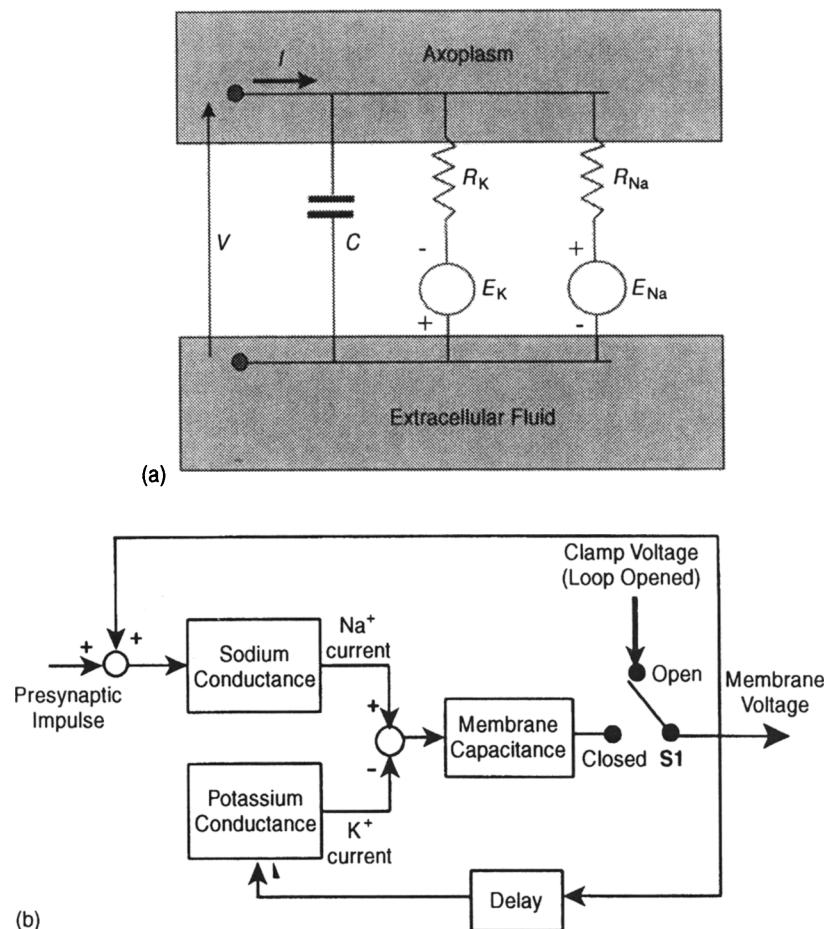
**Figure 7.11** Schematic representation of the artificial brain perfusion setup for separating central and peripheral chemoreflex drives.

#### 7.4.4 The Voltage Clamp

The basic mechanism for the neuronal action potential is a classic example of a physiological process in which both negative and positive feedback occur. Consider the Hodgkin–Huxley model shown in circuit form in Figure 7.12a and block diagram form in Figure 7.12b; the model has been simplified here to exclude the leakage channel due to the chloride ions. Normally, potassium ions ( $K^+$ ) tend to leak out of the nerve cell because of the much larger  $K^+$  concentration in the axoplasm relative to the extracellular fluid. The opposite occurs with the sodium ions ( $Na^+$ ). When the membrane is depolarized by a presynaptic stimulus, the variable  $Na^+$  conductance increases rapidly and considerably, allowing a large influx of  $Na^+$  ions from the extracellular fluid, which depolarizes the cell membrane even further. Thus, the positive feedback dominates this initial phase of the action potential. Fortunately, the increase in  $Na^+$  conductance is short-lived and the influx of  $Na^+$  ions slows after a fraction of a millisecond. At the same time, the  $K^+$  conductance starts to increase, following a short delay. This allows  $K^+$  ions to flow out of the axoplasm, acting to reverse the depolarization of the membrane. The repolarization speeds up the decline in  $Na^+$  conductance which, in turn, promotes the repolarization process (Figure 7.12b). Through the insertion of an electrode into the axoplasm, it is possible to control precisely the voltage inside the nerve cell. By applying a step depolarization through this electrode and keeping the applied voltage constant, one is effectively “opening” both the positive and negative feedback loops. Then, by measuring the current flowing across the membrane and by altering the composition of the extracellular fluid to isolate the  $Na^+$  from  $K^+$  effects, it is possible to deduce the time-courses of the  $Na^+$  and  $K^+$  conductances to the step depolarization. This was the basic methodology employed by Hodgkin, Huxley, and Katz (1952), as well as researchers after them, to study the mechanisms underlying the generation of the action potential.

#### 7.4.5 Opening the Pupillary Reflex Loop

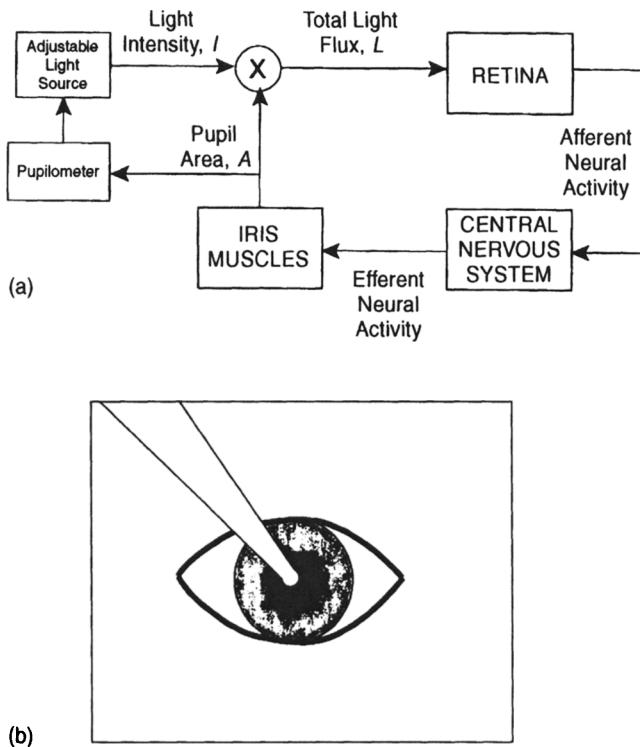
The model of the pupillary light reflex that we employed in Section 6.6 to demonstrate stability analysis was based largely on Stark’s ingenious experiments in which he developed techniques to functionally open the reflex loop. The two basic means by which this was done are illustrated in Figure 7.13. In the normal closed-loop state, an increase in total light flux impinging on the retina results in a reduction in pupil area which, assuming the light intensity remains constant, would decrease total light flux to offset the initial increase. In Figure 7.13a, Stark used a pupillometer to measure the size of the pupil and an adjustable light source that delivered light at intensities that were inversely proportional to the pupil area. By introducing these devices into the feedback loop, he was able to offset the effect of changing pupil area on total light flux by raising the light intensity. In this way, the total light flux could be controlled quite precisely, enabling him to deduce the loop transfer function characteristics of the reflex. Another technique that he used to effectively “open the loop” is illustrated in Figure 7.13b. Here, he applied a very narrow beam of light through the pupil. By restricting the cross-sectional area of the beam to a size that was smaller than the residual area of the pupil, total light flux was rendered completely independent of pupil area, since the area of the light beam impinging on the retina was not affected by changes in pupil size. Under these conditions, it was possible to completely control the time-course of the input (total light flux) and measure the corresponding response (pupil area) of the “opened” reflex loop.



**Figure 7.12** (a) Simplified schematic of the Hodgkin–Huxley model. (b) “Opening the loop” via application of the voltage clamp technique.

#### 7.4.6 Read Rebreathing Technique

Under normal operating circumstances, ventilation ( $\dot{V}_E$ ) and arterial  $P_{CO_2}$  ( $P_{aCO_2}$ ) are tightly coupled through the powerful negative feedback loops of the chemoreflexes: any increases in  $P_{aCO_2}$  lead rapidly to increases in  $\dot{V}_E$  that act to offset the initial rise in  $P_{aCO_2}$ . However, Read (1967) found a simple experimental technique of functionally breaking this closed-loop relationship. The subject breathes into and out of a small (4–6 liters) rebreathing bag which is filled with an initial gas mixture containing 7%  $CO_2$  in oxygen. After an initial transient phase, an equilibrium is established between arterial blood, oxygenated mixed venous blood, and gas in the lungs and rebreathing bag. Thereafter, the  $P_{CO_2}$  in both blood and gas phases increases linearly with time, and  $\dot{V}_E$  also increases proportionally, without reversing the rise in  $P_{CO_2}$  as one would expect in the closed-loop situation. The way in which

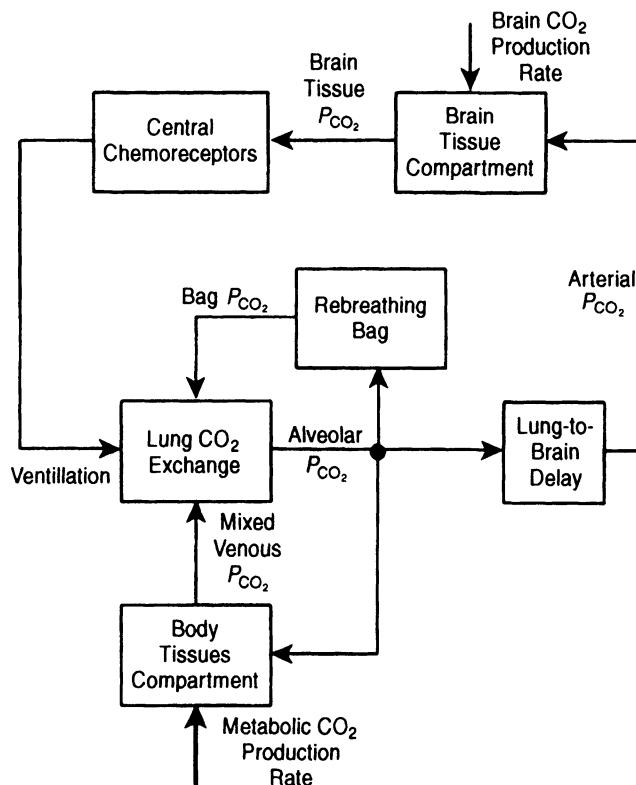


**Figure 7.13** Two methods of functionally opening the pupillary reflex loop: (a) modulation of applied light intensity using measurements of pupil area; (b) application of narrow light beam to residual area of pupil.

the technique works is best appreciated from a modeling perspective. The following differential equation provides the simplest dynamic characterization of  $\text{CO}_2$  exchange at the level of the body tissues:

$$\frac{V_t}{863} \frac{dP_{\text{vCO}_2}}{dt} = QK_{\text{CO}_2}(P_{\text{aCO}_2} - P_{\text{vCO}_2}) + \dot{V}_{\text{CO}_2} \quad (7.55)$$

Equation (7.55) assumes the capacitance effect of the body tissues to be lumped into the volume  $V_t$ .  $V_{\text{CO}_2}$  is the metabolic production rate of  $\text{CO}_2$ . Following the establishment of the equilibrium between the arterial and mixed venous blood and gas in the lungs and bag, it can be seen that in Equation (7.55) the arteriovenous gradient disappears and the derivative becomes a constant proportional to the  $\text{CO}_2$  metabolic production rate. Integrating Equation (7.55) results in  $P_{\text{vCO}_2}$  assuming a linear dependence on time. Since  $P_{\text{aCO}_2}$ , alveolar  $P_{\text{CO}_2}$  ( $P_{\text{ACO}_2}$ ) and the bag  $P_{\text{CO}_2}$  are equilibrated with  $P_{\text{vCO}_2}$ , these variables also increase linearly with time during the rest of the rebreathing process. With the linearly rising arterial and tissue  $P_{\text{CO}_2}$ , brain tissue  $P_{\text{CO}_2}$  will also increase in linear fashion, driving  $\dot{V}_E$  along a similar time-course. The increasing  $\dot{V}_E$  is mediated almost completely by the central chemoreceptors, since the high oxygenation levels suppress peripheral chemoreception. However, because of the equilibration between the bag (inspired)  $P_{\text{CO}_2}$  and  $P_{\text{ACO}_2}$  (see Equation (6.42) in Section 6.7), the increasing  $\dot{V}_E$  is prevented from influencing  $P_{\text{ACO}_2}$ , hence breaking the negative



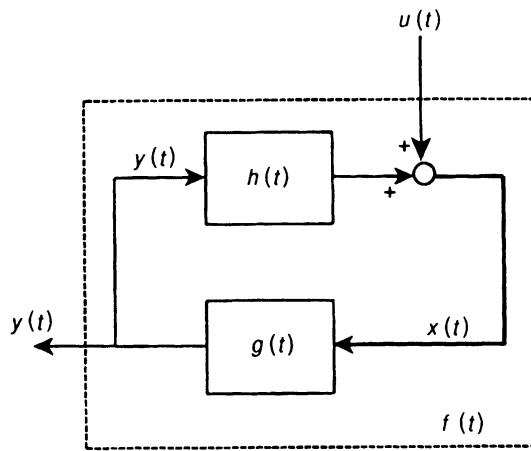
**Figure 7.14** Schematic block diagram of respiratory control during rebreathing.

feedback in this closed-loop system. A schematic block diagram of the rebreathing model is shown in Figure 7.14.

## 7.5 IDENTIFICATION UNDER CLOSED-LOOP CONDITIONS: CASE STUDIES

Although we have seen a wide range of physiological examples in which experimental interventions were employed to “open the loop,” such techniques are not always applicable. Moreover, a major criticism leveled against this kind of approach is that the system under study is placed under nonphysiological conditions and subjected to nonphysiological inputs when these interventions are applied. Ideally, we would like to identify the physiological system under “normal operating conditions” when its feedback loops are functionally intact. However, consider the problem involved with identifying the impulse response  $h(t)$  of the closed-loop system component shown in Figure 7.15. The unknown (and unobservable) disturbance,  $u(t)$ , that enters the closed loop represents both a “measurement” and “process” noise input. It is considered “measurement noise” since it corrupts the measurements  $x(t)$ , which otherwise would be related solely to  $y(t)$ , the input to the system component. This is clear from the mathematical expression relating  $x(t)$  to  $y(t)$  and  $u(t)$ :

$$x(t) = \int_0^{\infty} h(\tau)y(t - \tau) dt + u(t) \quad (7.56)$$



**Figure 7.15** Correlation of the process noise input,  $u(t)$ , with  $y(t)$  complicates the identification  $h(t)$  from closed-loop measurements,  $y(t)$  and  $x(t)$ .  $f(t)$  is the impulse response of the closed-loop system.

$u(t)$  also takes the form of “process noise” since it enters the closed-loop system and becomes correlated with  $y(t)$ . If we consider  $u(t)$  the input and  $y(t)$  the output of the overall system (as defined by the dashed rectangle in Figure 7.15), we obtain

$$y(t) = \int_0^{\infty} f(\tau)u(t - \tau) d\tau \quad (7.57)$$

In order to obtain an unbiased estimate of  $h(t)$  from Equation (7.56) by the least squares approach using  $y(t)$  as input and  $x(t)$  as output, the final solution must be such that  $u(t)$  becomes uncorrelated with (or orthogonal to)  $y(t)$ . However, as Equation (7.57) clearly shows,  $y(t)$  is correlated with  $u(t)$ . Thus, the direct application of open-loop system identification methods to this problem will not yield accurate estimates of  $h(t)$ . A couple of approaches for circumventing this problem are described in the following two subsections. One other approach is to impose constraints on the orthogonality condition, but this falls outside the scope of the present discussion. For further information about this last method, the interested reader is referred to a study by Khoo (1989).

### 7.5.1 Minimal Model of Blood Glucose Regulation

One effective way of partitioning the effects of the feedforward and feedback components of a closed-loop system from one another is to assume a model structure for at least one of these components. Then, if the effects of all other extraneous influences (process noise) entering the closed-loop system are small relative to the magnitude of the system responses, the parameters of the assumed model can be estimated. The “minimal model” of blood glucose regulation, developed by Bergman and colleagues (1979), represents a good example of this kind of approach. Referring to Figure 7.15, suppose that  $x(t)$  and  $y(t)$  correspond to the plasma glucose and insulin concentrations at time  $t$ , respectively. Then, the impulse response function,  $h(t)$ , would represent glucose regulation kinetics, while  $g(t)$  would reflect the dynamics of insulin production and utilization. The closed-loop system is perturbed by an impulsive input  $u(t)$ , consisting of an intravenous injection ( $300 \text{ mg kg}^{-1}$  in dogs) of glucose. By using the resulting time-courses in  $y(t)$  and  $x(t)$  as *input* and *output*, respectively, the model of *glucose dynamics* can be identified. Subsequently, by using  $x(t)$  as *input* and  $y(t)$  as *output*, the parameters of the model of *insulin dynamics* can be estimated. Bergman and coworkers have referred to this methodology as *partition analysis*, since both

halves of the closed-loop system are identified as if they were in the open-loop state. It should be emphasized that the key assumptions that make this kind of closed-loop estimation possible are: (a) the imposition of structure and causality on the dynamics characterizing glucose and insulin production and utilization; and (b) relatively large signal-to-noise ratios in the measurements.

In this section, we will discuss only the estimation of the minimal model of glucose regulation, i.e., how insulin affects glucose. The estimation of the converse model in which glucose affects insulin will not be considered. Thus, the input here is the measured plasma insulin concentration  $y(t)$  following the intravenous glucose injection, while the output is the corresponding measured blood glucose concentration  $x(t)$ . The model employed by Bergman contains the features incorporated in the glucose kinetics model proposed by Stolwijk and Hardy (see Sections 3.6 and 5.5), but is more realistic in that it allows for the delayed effect of insulin on glucose disappearance, a feature that has been observed. The insulin concentration,  $y(t)$ , does not affect glucose dynamics directly. Instead, it acts through a “remote compartment,” so that the *effective* insulin concentration,  $y_{\text{eff}}(t)$  is given by

$$\frac{dy_{\text{eff}}}{dt} = k_2 y(t) - k_3 y_{\text{eff}}(t) \quad (7.58a)$$

where  $k_2$  and  $k_3$  represent the fractional rate parameters for insulin transport into and elimination from the remote compartment. This compartment is “remote” in that  $y_{\text{eff}}$  is not directly measurable. It should also be noted that the volume of the remote compartment has been factored into the rate constants  $k_2$  and  $k_3$ . The rate of change of glucose in the blood plasma is given by

$$\frac{dx}{dt} = \frac{\text{Net rate of glucose production by the liver}}{\text{Rate of glucose utilization by other tissues}} - \frac{\text{Rate of glucose utilization by other tissues}}{\text{Rate of glucose utilization by other tissues}} \quad (7.59a)$$

where

$$\text{Net rate of glucose production by the liver} = B_0 - k_5 x(t) - k_6 y_{\text{eff}}(t)x(t) \quad (7.60)$$

and

$$\text{Rate of glucose utilization by other tissues} = R_{d0} + k_1 x(t) + k_4 y_{\text{eff}}(t)x(t) \quad (7.61)$$

In Equation (7.60),  $B_0$  represents the rate of glucose production by the liver. The rate of glucose uptake by the liver is assumed to be proportional to an insulin-independent component (through rate constant  $k_5$ ) and an insulin-dependent component (through rate constant  $k_6$ ). Similarly, in Equation (7.61), the rate of glucose utilization by nonhepatic tissues is assumed to have a constant component, a component proportional to glucose concentration and a component sensitive to both glucose and effective insulin concentration. Substituting Equations (7.60) and (7.61) into Equation (7.59a) and rearranging terms, we obtain the result:

$$\frac{dx}{dt} = [B_0 - R_{d0}] - [k_5 + k_1]x(t) - [k_6 + k_4]y_{\text{eff}}(t)x(t) \quad (7.59b)$$

As in Equation (7.58a), the effective plasma glucose capacitance is factored into the parameters on the right-hand side of Equation (7.59b).

Equations (7.58a) and (7.59b) provide a complete characterization of glucose kinetics. However, it is obvious that there are too many redundant parameters. For instance, in Equation (7.59b), it would not be possible to estimate  $B_0$  and  $R_{d0}$  separately; only the combined term  $[B_0 - R_{d0}]$  can be identified. The same is true for  $[k_5 + k_1]$  and  $[k_6 + k_4]$ . In

addition, since  $y_{\text{eff}}(t)$  is not measurable, a further reduction in parametrization can be achieved by defining the new variable  $z(t)$  that is proportional to  $y_{\text{eff}}(t)$ :

$$z(t) \equiv [k_6 + k_4]y_{\text{eff}}(t) \quad (7.62)$$

Substituting Equation (7.62) into Equations (7.58a) and (7.59b) we obtain

$$\frac{dz}{dt} = -p_2 z(t) + p_3 y(t) \quad (7.58b)$$

and

$$\frac{dx}{dt} = p_4 - p_1 x(t) - z(t)x(t) \quad (7.59c)$$

where  $p_1 = k_1 + k_5$ ,  $p_2 = k_3$ ,  $p_3 = k_2(k_4 + k_6)$ , and  $p_4 = B_0 - R_{d0}$ . Equations (7.58b) and (7.59c) provide the same dynamic characterization of glucose regulation for the minimum number of unknown parameters that have to be estimated from the input–output data. For this reason, it is referred to as a *minimal model*.

The way in which the unknown parameters  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$  are estimated is as follows. First, we begin with initial guesses for the unknown parameters. Using the measured input time-course,  $y(t)$ , and the initial parameter values, Equation (7.58b) is first solved to obtain the value of  $z$  at the current time-step. Using this value of  $z$  in Equation (7.59c) and integrating this equation, the glucose concentration at the next time step can be computed. This process is repeated until predictions for  $x(t)$  have been made for the entire duration of the experiment. The predictions are compared to the actual blood glucose measurements, and the value of the criterion function (sum of squares of the differences between measured and predicted glucose values) is computed. An optimization algorithm is used to search for another combination of the four unknown parameters that would produce a lower value of the criterion function. Using the new combination of parameter values,  $z(t)$  and  $x(t)$  are again solved using Equations (7.58b) and (7.59c), and the whole process is repeated until the incremental reduction in criterion function is considered insignificant.

An example of the results achieved with minimal model estimation is displayed in Figure 7.16. “Data” required for the estimation were generated using a SIMULINK implementation (named “gmm\_sim.mdl”) of Bergman’s models of both glucose and insulin subsystems (Bergman et al., 1979; Bergman et al., 1985; Toffolo et al., 1980). These were combined and made to operate in closed-loop mode. Random perturbations were added to the glucose concentration,  $x(t)$ , predicted by the model to simulate “measurement noise” in the glucose observations. The SIMULINK model, shown in Figure 7.17, produced samples of  $x(t)$  and the plasma insulin concentration,  $y(t)$ , at intervals of 1 minute to mimic the blood sampling conducted in the real experiments. These “measurements” are shown as the solid circles in both upper and lower panels of Figure 7.16. This particular set of “measurements” has also been saved in the MATLAB data file: “data\_gmm.mat”. Parameter estimation is performed using the Nelder–Mead simplex algorithm, which is implemented in MATLAB with the function “fmins”. The primary command lines of the MATLAB script file, labeled “gmm\_est.m”, are:

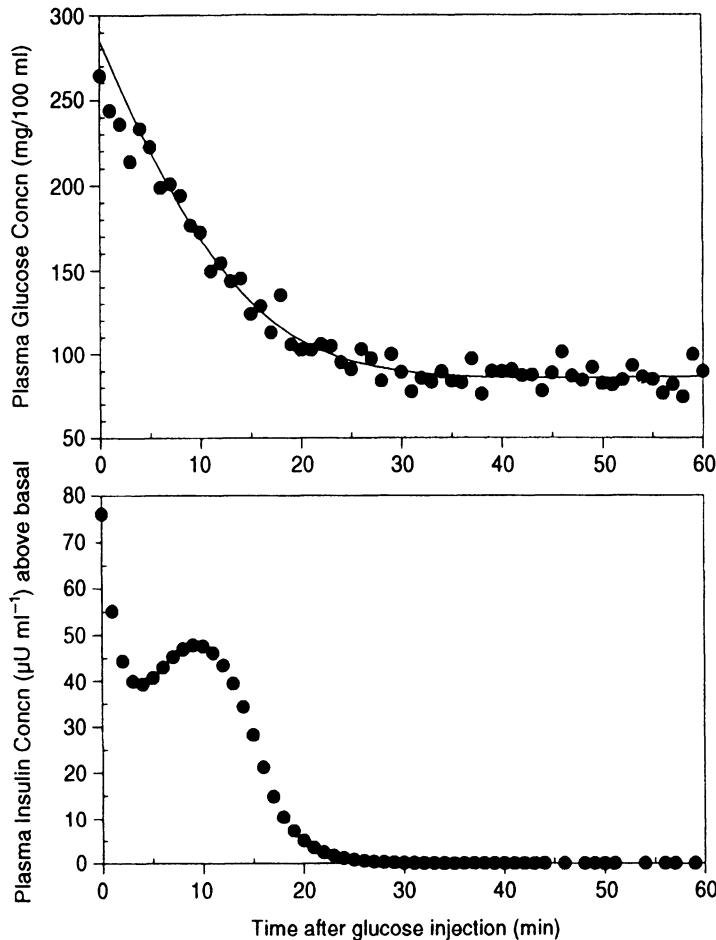
```
>> options(1)=1;
>> [p,options] = fmins('fn_gmm',p_init,options,[]);
```

where the function “fn\_gmm” (in MATLAB file “fn\_gmm.m”) is used by “gmm\_est.m” to produce values of the criterion function  $J$  with each iteration of the algorithm. For each new set of parameter values, “fn\_gmm” solves the model equations given in Equations (7.58b) and (7.59c) using the Euler method of integration (with time-steps of 0.01 minute) and computes the sum of squares of the differences between the “observed” glucose concentration samples and the values predicted by model solution. The parameters to be estimated are the four unknown coefficients in Equations (7.58b) and (7.59c):  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$ ; in addition, the “true” glucose concentration at time zero,  $x(0)$ , is treated as the fifth unknown parameter. In the above MATLAB command lines, note that the array “options” is used as both an input and output argument. This is done (specifically, the first element of “options” is set equal to “1”) to allow the algorithm to display the value of  $J$ , along with the parameter values associated with the simplex vertices, at each stage of the computations. The final estimated set of parameter values are  $p_1 = 0.068$ ,  $p_2 = 0.091$ ,  $p_3 = 6.72 \times 10^{-5}$ ,  $p_4 = 6.03$  and  $x(0) = 284.9$  mg/100 ml. These may be compared with the “true” parameter values used in the SIMULINK program:  $p_1 = 0.049$ ,  $p_2 = 0.091$ ,  $p_3 = 8.96 \times 10^{-5}$ , and  $p_4 = 4.42$ . These latter values were selected from the results obtained by Bergman and coworkers from their experiments on dogs. The “best-fit” model prediction is shown as the solid curve in Figure 7.16.

The minimal model has been employed successfully in many clinical studies to quantitate insulin sensitivity and glucose effectiveness in various populations at risk for diabetes. More details on the relationship between these indices and the model parameters may be found in the original papers by Bergman and coworkers listed in the Bibliography section of this chapter.

### 7.5.2 Closed-Loop Identification of the Respiratory Control System

In this section, we illustrate a somewhat different approach to closed-loop identification. In the previous example, an optimization technique was employed for parameter estimation. As we mentioned earlier, one disadvantage of this kind of iterative method is the possibility of convergence to a local minimum instead of the global solution. Here, we take the alternative approach of least squares estimation, where the optimal solution is arrived at in one computational step. Another difference that we will highlight here is the use of a *persistently exciting* input to stimulate the closed-loop system, instead of the brief but potent impulsive disturbance employed in the minimal model of glucose regulation. Practical considerations dictate the use of the former type of input in the case of the respiratory control system. A potent impulsive disturbance in this case would take the form of an inhaled breath of gas with very high  $\text{CO}_2$  content. Such a potent stimulus would be certain to evoke a behavioral response in addition to the chemoreflex-mediated changes, thereby allowing the measurement process to affect the system under observation. In this case, the stimulus takes the form of a pseudorandom binary sequence (PRBS) in the inhaled  $P_{\text{CO}_2}$ . This allows the system to be excited with relatively low  $\text{CO}_2$  concentrations over a broad range of frequencies within the limited experimental duration. An example of the practical implementation of this kind of PRBS time-course in inhaled  $P_{\text{CO}_2}$  ( $P_{\text{ICO}_2}$ ) and the resulting effects on alveolar  $P_{\text{CO}_2}$  ( $P_{\text{ACO}_2}$ ) and ventilation ( $\dot{V}_{\text{E}}$ ) in a normal human subject is displayed in Figure 7.18. As in the previous section, partition analysis is employed in the identification procedure. The first stage of the analysis involves the estimation of the parameters of the plant (i.e., gas exchange in the

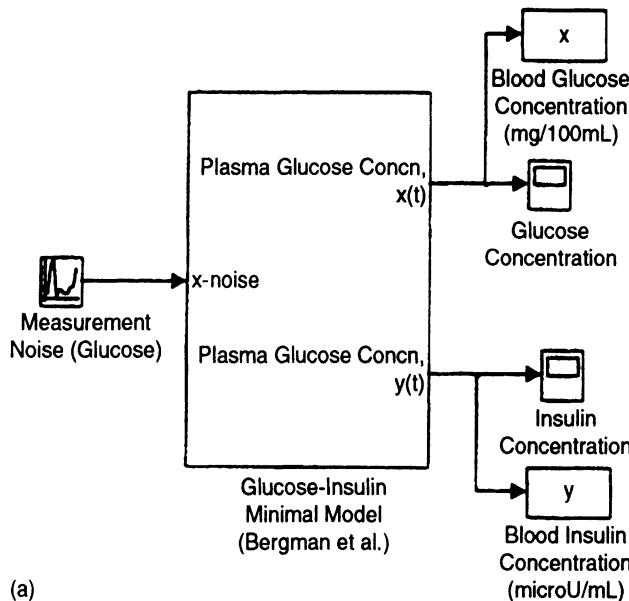


**Figure 7.16** “Measurements” of blood glucose and insulin levels (solid circles) following intravenous bolus infusion of  $300 \text{ mg kg}^{-1}$  of glucose. The best-fit prediction for glucose is shown as the solid curve.

lungs), using measurements of  $\dot{V}_E$  and  $P_{\text{ICO}_2}$  as inputs, and  $P_{\text{ACO}_2}$  as output. The second stage consists of the estimation of the controller and lung-to-chemoreceptor delay using  $P_{\text{ACO}_2}$  as the input and  $\dot{V}_E$  as the output (Figure 7.19).

**7.5.2.1 Identification of the plant.** The model employed to represent the CO<sub>2</sub> exchange in the lungs is the small-signal expression derived in Equation (6.43b) in Section 6.7.1. However, in the expression shown below, we have also allowed for perturbations in  $P_{\text{ICO}_2}$  (which, in Equation (6.43b), was kept constant)

$$\tau_{\text{lung}} \frac{d(\Delta P_{\text{ACO}_2})}{dt} + \Delta P_{\text{ACO}_2 = G_1 \Delta P_{\text{ICO}_2} - G_2 \Delta \dot{V}_E} \quad (7.63)$$



**Figure 7.17** SIMULINK model “gmm\_sim.mdl” of combined glucose–insulin kinetics based on Bergman’s minimal models: (a) model input and outputs.

where  $\Delta P_{\text{ACO}_2}$ ,  $\Delta \dot{V}_E$  and  $\Delta P_{\text{ICO}_2}$  represent small changes in  $P_{\text{ACO}_2}$  (assumed equal to  $P_{\text{aCO}_2}$ ),  $\dot{V}_E$ , and  $P_{\text{ICO}_2}$  about their equilibrium values, and

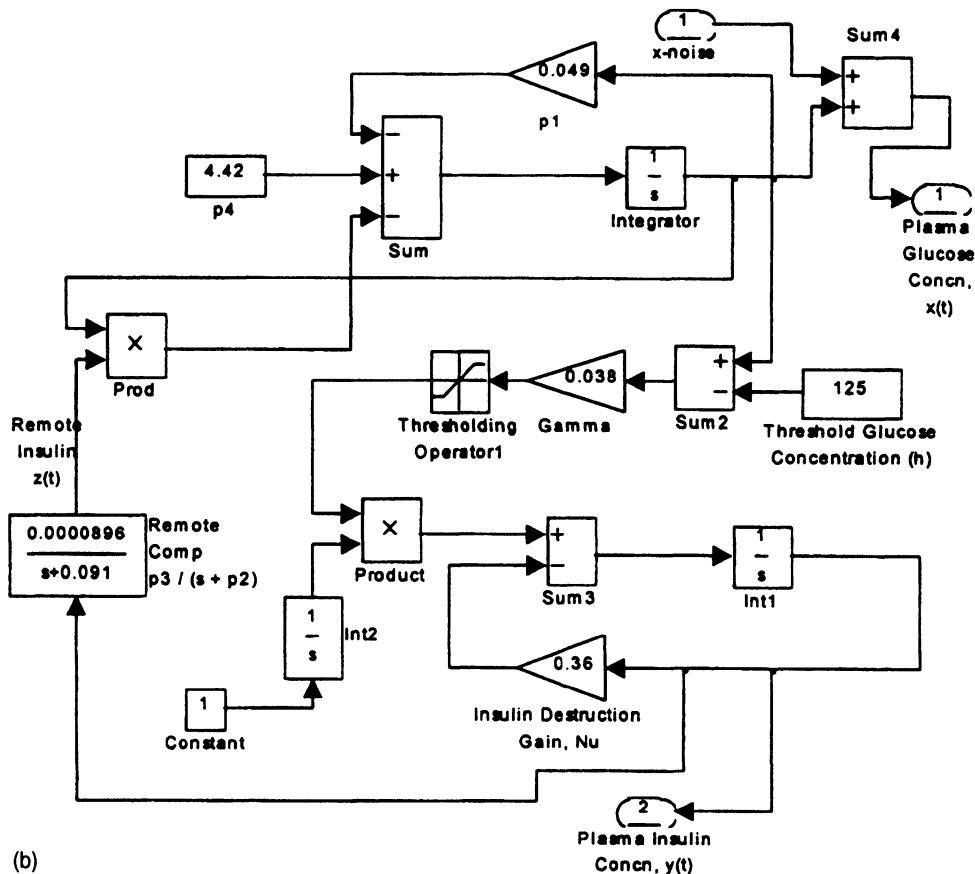
$$G_1 = \frac{\dot{V}_E - \dot{V}_D}{\dot{V}_E - \dot{V}_D + 863QK_{\text{CO}_2}} \quad (7.64)$$

$$G_2 = \frac{P_{\text{ACO}_2} - P_{\text{ICO}_2}}{\dot{V}_E - \dot{V}_D + 863QK_{\text{CO}_2}} \quad (7.65)$$

Since, in this analysis, we are limiting our attention to the characterization of how small changes in  $P_{\text{ICO}_2}$  elicit changes in  $P_{\text{ACO}_2}$  and  $\dot{V}_E$ , we assume, to a first approximation, that the operating values of  $P_{\text{ACO}_2}$ ,  $\dot{V}_E$ ,  $\dot{V}_D$ , and  $P_{\text{ICO}_2}$  are constant. Hence, we regard the two factors  $G_1$  and  $G_2$  to be constant-valued parameters, which have to be estimated from the measurements, as we demonstrate below.

Since the measurements of  $P_{\text{ACO}_2}$ ,  $P_{\text{ICO}_2}$  and  $\dot{V}_E$  are not made continuously in time but are obtained on a breath-by-breath basis, for purposes of parameter estimation it is more useful to express the plant equation in the form of a difference equation with a discrete-time base (with “breaths” as the unit of time). Furthermore,  $P_{\text{ACO}_2}$  cannot be directly sampled; instead, we assume that end-tidal  $P_{\text{CO}_2}$  (the highest value of  $P_{\text{CO}_2}$  measured in the exhaled stream during expiration) reliably reflects  $P_{\text{ACO}_2}$ . By integrating Equation (7.63) from the end of the previous breath to the end of the current breath, the differential equation can be converted into a difference of the following form:

$$\Delta P_{\text{ACO}_2}(n) + \alpha \Delta P_{\text{ACO}_2}(n-1) = \beta_1 \Delta P_{\text{ICO}_2}(n) - \beta_2 \Delta \dot{V}_E(n) + e(n) \quad (7.66)$$



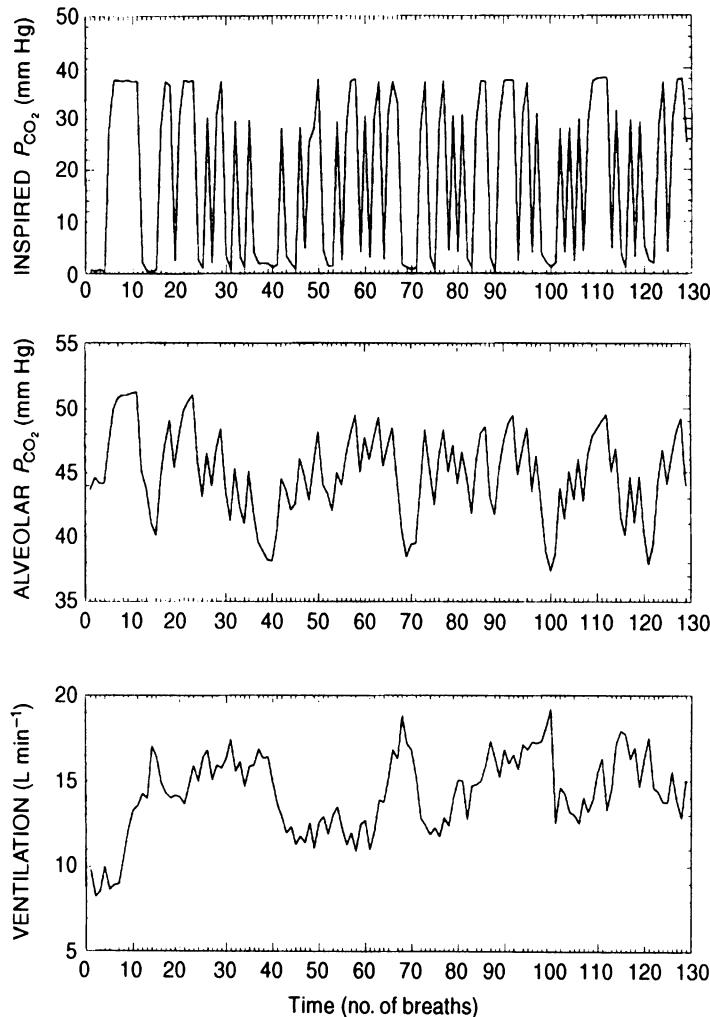
**Figure 7.17** SIMULINK model “gmm\_sim.mdl” of combined glucose–insulin kinetics based on Bergman’s minimal models: (b) details of the SIMULINK implementation.

where  $n$  represents the current breath number, and  $0 \leq n \leq N - 1$ ,  $N$  being the total number of breaths used for data analysis. This type of continuous-time to discrete-time conversion is also known as the *impulse invariance* method; further details on this method may be found in most texts on signal processing, e.g., Jackson (1995). It can be further shown that  $\alpha$ ,  $\beta_1$ , and  $\beta_2$  in Equation (7.66) are related to the parameters  $G_1$ ,  $G_2$ , and  $\tau_{\text{lung}}$  of Equation (7.63) through the following relations:

$$\beta_1 = \frac{G_1}{\tau_{\text{lung}}} \quad (7.67)$$

$$\beta_2 = \frac{G_2}{\tau_{\text{lung}}} \quad (7.68)$$

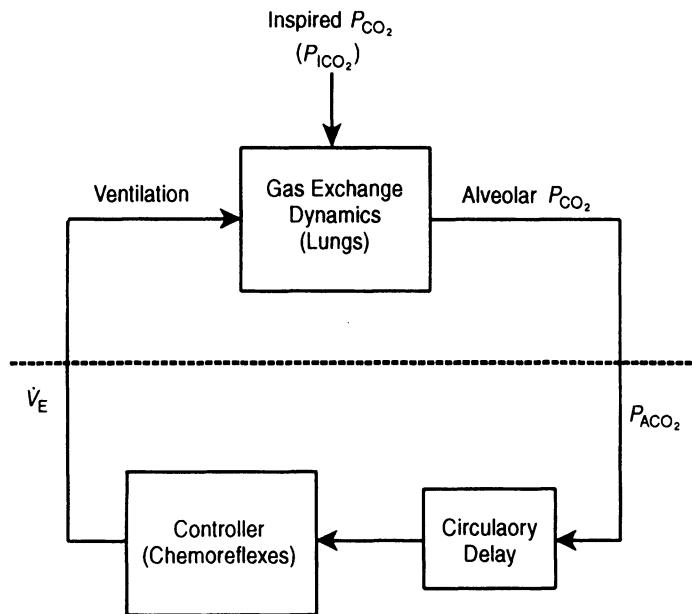
$$\alpha = -e^{-T/\tau_{\text{lung}}} \quad (7.69)$$



**Figure 7.18** Responses in  $P_{ACO_2}$  (middle panel) and ventilation (bottom panel) produced in a normal subject during inhalation of 6%  $CO_2$  in air, administered on a pseudorandom binary basis ( $P_{ICO_2}$ , top panel). Reproduced from Ghazan-shahi and Khoo (1997).

where  $T$  is the “sampling interval” which, in this case, would be the breath duration. Strictly speaking,  $T$  would vary from breath to breath, since the breathing frequency is somewhat variable. However, previous studies in this field have demonstrated that assuming  $T$  to be constant and equal to the *average breath duration* simplifies matters considerably without affecting the outcome of the analysis significantly in most experimental situations.

In Equation (7.66), the last term  $e(n)$  is added to account for the residual error between the measured  $\Delta P_{ACO_2}$  and model-predicted  $\Delta P_{ACO_2}$ . Equation (7.66) is a special case of the general class of models known as *ARX* (autoregressive with exogenous input) models (Ljung, 1987). In this special case,  $\alpha$ ,  $\beta_1$ , and  $\beta_2$  are the unknown parameters to be estimated using  $\Delta P_{ACO_2}$  as the output measurement, and  $\Delta P_{ICO_2}$  and  $\Delta V_E$  as the inputs. Estimation of these



**Figure 7.19** Application of partition analysis for identification of the plant (above dashed line) and controller (below dashed line) portions of the closed loop.

parameters can be easily achieved using least squares minimization, as the following equations illustrate. Rewriting Equation (7.66) for all values of  $n$  in vector form, we have

$$\begin{bmatrix} \Delta P_{ACO_2}(0) \\ \Delta P_{ACO_2}(1) \\ \vdots \\ \Delta P_{ACO_2}(N-1) \end{bmatrix} = \begin{bmatrix} 0 & \Delta P_{ICO_2}(0) & -\Delta \dot{V}_E(0) \\ -\Delta P_{ACO_2}(0) & \Delta P_{ICO_2}(1) & -\Delta \dot{V}_E(1) \\ \dots & \dots & \dots \\ -\Delta P_{ACO_2}(N-2) & \Delta P_{ICO_2}(N-1) & -\Delta \dot{V}_E(N-1) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(N-1) \end{bmatrix} \quad (7.70)$$

It can be seen that Equation (7.70) is of the form displayed in Equation (7.8), i.e.,

$$\underline{\mathbf{y}} = \mathbf{U}\underline{\mathbf{h}} + \underline{\mathbf{e}} \quad (7.71)$$

where  $\underline{\mathbf{y}}$  represents the column vector on the left-hand side of Equation (7.70),  $\mathbf{U}$  is the  $N \times 3$  matrix on the other side of the equation, and  $\underline{\mathbf{h}}$  is the vector containing the unknown parameters. Thus,  $\underline{\mathbf{h}}$  can be estimated using Equation (7.12), which we rewrite below:

$$\underline{\mathbf{h}} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\underline{\mathbf{y}} \quad (7.72)$$

**7.5.2.2 Identification of the Controller and Circulatory Delay.** To model the controller and circulatory delay, we assume the form proposed by Bellville et al. (1979)

(see Section 6.7.3). Using the Laplace transform version of this model, we have (from Equation (6.50a,b)):

$$\Delta \dot{V}_E(s) = \left( \frac{G_c}{\tau_c s + 1} + \frac{G_p}{\tau_p s + 1} \right) e^{-sT_d} \Delta P_{ACO_2}(s) \quad (7.73a)$$

Here, we have made the simplifying assumption of using only one common circulatory delay,  $T_d$ , in place of the separate central ( $T_c$ ) and peripheral ( $T_p$ ) delays assumed in the chemoreflex model of Section 6.7.3. Employing a common denominator for both terms in the summation of Equation (7.73a), we can rewrite the equation in the following form:

$$\Delta \dot{V}_E(s) = \left( \frac{(G_c \tau_p + G_p \tau_c)s + (G_c + G_p)}{\tau_c \tau_p s^2 + (\tau_c + \tau_p)s + 1} \right) e^{-sT_d} \Delta P_{ACO_2}(s) \quad (7.73b)$$

When inverse-Laplace-transformed back into the time-domain, Equation (7.73b) takes the following differential equation form:

$$\begin{aligned} \tau_c \tau_p \frac{d^2(\Delta \dot{V}_E)}{dt^2} + (\tau_c + \tau_p) \frac{d(\Delta \dot{V}_E)}{dt} + \Delta \dot{V}_E &= (G_c \tau_p + G_p \tau_c) \frac{d(\Delta P_{ACO_2}(t - T_d))}{dt} \\ &\quad + (G_c + G_p) \Delta P_{ACO_2}(t - t_d) \end{aligned} \quad (7.74)$$

As in Section 7.5.2.1, since the measurements are made on a breath-by-breath basis, it is more convenient to assume a discrete-time base and recast the model in finite difference form, as in Equation (7.66) for the plant. In this case, the corresponding finite difference equation is

$$\begin{aligned} \Delta \dot{V}_E(n) + a_1 \Delta \dot{V}_E(n-1) + a_2 \Delta \dot{V}_E(n-2) \\ = b_0 \Delta P_{ACO_2}(n - N_d) + b_1 \Delta P_{ACO_2}(n-1 - N_d) + \epsilon(n) \end{aligned} \quad (7.75)$$

where, as in Equation (7.66),  $n$  represents the current breath number and  $0 \leq n \leq N - 1$ .  $N_d$  represents the circulatory delay in number of breaths, i.e.,  $N_d = T_d/T$ . In this case,  $\epsilon(n)$  is added to account for the discrepancy between the model-predicted  $\Delta \dot{V}_E$  and the measured  $\Delta \dot{V}_E$ . Equation (7.75) can also be cast in the form

$$\Delta \dot{V}_E = \begin{bmatrix} -\Delta \dot{V}_E(n-1) & -\Delta \dot{V}_E(n-2) & \Delta P_{ACO_2}(n - N_d) & \Delta P_{ACO_2}(n - N_d - 1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_0 \\ b_1 \end{bmatrix} + \epsilon(n) \quad (7.76)$$

By applying Equation (7.76) to all  $N$  sets of data points, we can again construct a matrix equation of the form displayed in Equation (7.70), and thus estimate the unknown parameters  $a_1$ ,  $a_2$ ,  $b_0$ , and  $b_1$  using least squares minimization. However, in order to solve for the unknown parameters, it is necessary to know what  $N_d$  is. Determination of  $N_d$  is done in the following way. We first select a range of physiologically feasible values for  $N_d$ . The lung-to-eye delay in most normals is generally in the range of 6 to 12 seconds. Thus, a reasonable range for  $N_d$  might be 1 to 4. For each of these values of  $N_d$ , we solve the least squares minimization problem and estimate  $a_1$ ,  $a_2$ ,  $b_0$ , and  $b_1$ . For each case, we compute  $J$ , the residual sum of squares of the differences between the measured and predicted  $\Delta \dot{V}_E$  (as given in Equation (7.35)). The “best” estimate of  $N_d$  is that value that yields the lowest value of  $J$ .

Having estimated the unknown parameters,  $a_1$ ,  $a_2$ ,  $b_0$ , and  $b_1$ , it is possible in principle to relate them to the gains ( $G_c$  and  $G_p$ ) and time constants ( $\tau_c$  and  $\tau_p$ ) that characterize the

corresponding differential equation (Equation (7.74)) in a way similar to Equations (7.67) through (7.69). However, in this case, the relations will be nonlinear and the latter group of parameters would generally tend to be sensitive to errors in the estimates of  $a_1$ ,  $a_2$ ,  $b_0$ , and  $b_1$ . A more robust alternative approach is to characterize the controller dynamics in terms of its corresponding unit impulse response. This can be achieved quite easily by setting  $\Delta P_{\text{ACO}_2}(0)$  to 1 and  $\Delta P_{\text{ACO}_2}(n)$  to 0 for all  $n > 0$ , and computing  $\Delta \dot{V}_{\text{E}}$  recursively from Equation (7.75). The error terms,  $\epsilon(n)$ , are set equal to zero during this computation. Note that this would produce results similar to the PRBS technique described in Section 7.3.3. However, one can expect much less noisy estimates of the impulse response function from the present method, since  $h(n)$  in this case would be derived from only four parameters estimated from a large number (128 or higher) of datapoints. Further details on the application of this approach to human respiratory data and the results obtained may be found in journal papers by Khoo et al. (1995) and Ghazanshahi and Khoo (1997). In the problems given at the end of this chapter, the reader will be able to explore this technique in greater detail by applying the accompanying MATLAB script file “rcs\_est.m” to a number of datasets (“prbs1.mat”, “prbs2.mat”, “prbs3.mat”, “prbs4.mat”) obtained from human experiments.

## BIBLIOGRAPHY

- Bekey, G.A. System identification: an introduction and a survey. *Simulation* **16**: 151–166, 1970.
- Bellville, J.W., B.J. Whipp, R.D. Kaufman, G.D. Swanson, K.A. Aqleh, and D.M. Wiberg. Central and peripheral chemoreflex loop gain in normal and carotid body-resected subjects. *J. Appl. Physiol.* **46**: 843–853, 1979.
- Bergman, R.N., Y. Ziya Ider, C.R. Bowden, and C. Cobelli. Quantitative estimation of insulin sensitivity. *Am. J. Physiol.* **236**: E667–E677, 1979.
- Bergman, R.N., D.T. Finegood, and M. Ader. Assessment of insulin sensitivity in vivo. *Endocrine Rev.* **6**: 45–86, 1985.
- Bergman, R.N., and J.C. Lovejoy (Eds.). *The Minimal Model Approach and Determinants of Glucose Tolerance*. Louisiana State University Press, Baton Rouge, LA, 1997.
- Berkenbosch, A., J. Heeringa, C.N. Olievier, and E.W. Kruyt. Artificial perfusion of the ponto-medullary region of cats: A method for separation of central and peripheral effects of chemical stimulation of ventilation. *Respir. Physiol.* **37**: 347–364, 1979.
- Ghazanshahi, S.D., and M.C.K. Khoo. Estimation of chemoreflex loop gain using pseudorandom binary  $\text{CO}_2$  stimulation. *IEEE Trans. Biomed. Eng.* **44**: 357–366, 1997.
- Hill, J.D., and G.J. McMurtry. An application of digital computers to linear system identification. *IEEE Trans. Auto. Cont.* **AC-9**: 536–538, 1964.
- Hodgkin, A.L., A.F. Huxley, and B. Katz. Measurement of current–voltage relations in the membrane of the giant axon of *Loligo*. *J. Physiol. (London)* **116**: 424–448, 1952.
- Jackson, L.B. *Digital Filters and Signal Processing*, 3d ed. Kluwer Academic, Boston, 1995.
- Kao, F.F., and L.H. Ray. Regulation of cardiac output in anesthetized dogs during induced muscular work. *Am. J. Physiol.* **179**: 255–260, 1954.
- Khoo, M.C.K. Noninvasive tracking of peripheral ventilatory response to  $\text{CO}_2$ . *Int. J. Biomed. Comput.* **24**: 283–295, 1989.
- Khoo, M.C.K. Estimation of chemoreflex gain from spontaneous breathing data. In: *Modeling and Parameter Estimation in Respiratory Control* (edited by M.C.K. Khoo), Plenum Press, New York, 1989; pp. 91–105.

- Khoo, M.C.K., F. Yang, J.W. Shin, and P.R. Westbrook. Estimation of dynamic chemoresponsiveness in wakefulness and non-rapid-eye-movement sleep. *J. Appl. Physiol.* **78**: 1052–1064, 1995.
- Korenberg, M.J., and I.W. Hunter. The identification of nonlinear biological systems: Wiener kernel approaches. *Ann. Biomed. Eng.* **18**: 629–654, 1990.
- Ljung, L. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- Marmarelis, P.Z., and V.Z. Marmarelis. *Analysis of Physiological Systems*. Plenum Press, New York, 1978.
- Patterson, S.W., H. Piper, and E.H. Starling. The regulation of the heart beat. *J. Physiol. (London)* **48**: 465, 1914.
- Read, D.J.C. A clinical method for assessing the ventilatory response to carbon dioxide. *Australas. Ann. Med.* **16**: 20–32, 1967.
- Schetzen, M. *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, New York, 1980.
- Sohrab, S., and S.M. Yamashiro. Pseudorandom testing of ventilatory response to inspired carbon dioxide in man. *J. Appl. Physiol.* **49**: 1000–1009, 1980.
- Stark, L. *Neurological Control Systems: Studies in Bioengineering*. Plenum Press, New York, 1968.
- Swanson, G.D. Biological signal conditioning for system identification. *Proc. IEEE* **65**: 735–740, 1977.
- Toffolo, G., R.N. Bergman, D.T. Finegood, C.R. Bowden, and C. Cobelli. Quantitative estimation of beta cell sensitivity to glucose in the intact organism: A minimal model of insulin kinetics in the dog. *Diabetes* **29**: 979–990, 1980.

## PROBLEMS

- P7.1.** One technique that has been used to assess lung mechanical function is known as the method of “forced oscillations.” In one variant of this method, a loudspeaker system is used to generate random pressure perturbations ( $P_{ao}$ , considered the “input”) that are directed into the subject’s airways. The resulting fluctuations in airflow ( $\dot{V}$ , considered the “output”) are measured. Using these input–output measurements, it is possible to deduce the quantities that represent airway resistance ( $R$ ), airway inertance ( $L$ ) and respiratory compliance ( $C$ ) by assuming a linear model of respiratory mechanics, such as the linear model that we have considered previously (see Figure 4.1). Using the MATLAB script file “`sensanl.m`,” perform a sensitivity analysis to assess parameter identifiability. Plot sensitivity curves such as those displayed in Figure 7.7, assuming nominal parameter values of  $R = 1.5 \text{ cm H}_2\text{O s}^{-1}$ ,  $L = 0.01 \text{ cm H}_2\text{O s}^2 \text{ L}^{-1}$ , and  $C = 0.1 \text{ L cm H}_2\text{O}^{-1}$ . Use the MATLAB “`randn`” function to generate the white noise sequence that represents the applied forcing in  $P_{ao}$ . Assume a time step of 0.01 s and a total duration of 25 s for each experimental trial. (Note: You will need to implement and solve the model differential equation in a function that will be called by “`sensanl.m`.”)
- P7.2.** The dataset provided in the file “`data_fo.mat`” contains measurements of the input (labeled “`Pao`”) and output (labeled “`Flow`”) signals measured during an application of the method of forced oscillations, described in Problem P7.1. Assuming the respiratory mechanics model structure shown in Figure 4.1, estimate the model parameters ( $R$ ,  $L$ , and  $C$ ) from the input–output data. Use the optimization technique discussed in Section 7.2.5.2. It is expected that you will modify and apply the MATLAB script file “`popt_llm.m`.” Perform the minimization using different starting parameter estimates in order to obtain several sets of final parameter estimates. The differences in values of each parameter will give you some idea of the estimation error.

- P7.3. Using the dataset in “`data_fo.mat`,” and assuming “`Pao`” to be the input and “`Flow`” to be the output, apply least squares estimation (see Section 7.2.2) to deduce the impulse response of the corresponding system. It is expected that you will modify the MATLAB script file “`sysid_ls.m`.” The sampling interval is 0.01 s. Assume the number of points in the impulse response to be 50. Compute also the error band associated with the impulse response estimate.
- P7.4. Use the SIMULINK model file “`gmm_sim.mdl`” to generate 10 sets of insulin–glucose “data”: in all cases, set the variance of the “measurement noise” at  $36 \text{ mg}^2/100 \text{ ml}^2$ , but in each case set the random generator seed to a different integer. For each dataset, use “`gmm_est.m`” to estimate the parameters  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$  of the minimal model. Then, from the results of all 10 datasets, compute the mean and standard error associated with each of the model parameters.
- P7.5. The datasets provided in the files “`prbs1.mat`,” “`prbs2.mat`,” “`prbs3.mat`,” and “`prbs4.mat`” represent measurements of  $P_{\text{ACO}_2}$  and  $\dot{V}_{\text{E}}$  obtained from four human subjects who were breathing from a gas mixture, the composition of which was alternated between air and 6%  $\text{CO}_2$  in air on a pseudorandom binary basis. One set of these measurements is displayed in Figure 7.18. Using the MATLAB script file “`rcs_est.m`,” estimate in each case: (a) the impulse response that characterizes the dynamics of gas exchange in the lungs; (b) the impulse response that characterizes the dynamics of the chemoreflexes; and (c) the lung-to-chemoreceptor delay. Assume the time-scale to be expressed in numbers of breaths. By applying the fast Fourier transform to the impulse responses in (a) and (b), deduce the corresponding frequency responses. How much intersubject variability is there in the responses?