CHAPTER 5

IDENTIFICATION OF LINEAR SYSTEMS

5.1 INTRODUCTION

This chapter will discuss methods for the identification of the linear system models described in Chapter 3. The objective is to lay the foundation for the discussion of nonlinear system identification methods to follow in Chapters 6–8, rather than to provide a comprehensive review of linear identification methods. Consequently, the focus will be on the identification of nonparametric, time domain models using a least-squares framework. Frequency domain and parametric methods will be dealt with only briefly.

5.1.1 Example: Identification of Human Joint Compliance

The identification methods discussed in this chapter will be illustrated using datasets simulated from a second-order, linear, low-pass model of human ankle compliance. Low-order linear models of this type have been used extensively to describe physiological systems, including the mechanics of joints (Kearney and Hunter, 1990), air flow through the lungs (Lutchen and Suki, 1996), and the subthreshold dynamics of neurons (D'Aguanno et al., 1986).

Figure 5.1 shows block diagrams of the three simulations that will be used:

1. The "ideal" case, Figure 5.1A, was generated by applying a Gaussian white input directly to the second-order system. The input and output were sampled at 500 Hz; Figures 5.2A and 5.2B show a 2-s segment of input and output data for this case.
2. The "broadband" case, Figure 5.1B, was generated by filtering the input and output from the "ideal" case with an eighth-order, 200-Hz, low-pass, Bessel filter, to represent the effects of anti-alias filtering. The middle row of Figures 5.2C and 5.2D illustrate a typical input–output pair for this case.

Identification of Nonlinear Physiological Systems, By David T. Westwick and Robert E. Kearney
ISBN 0-471-27456-9

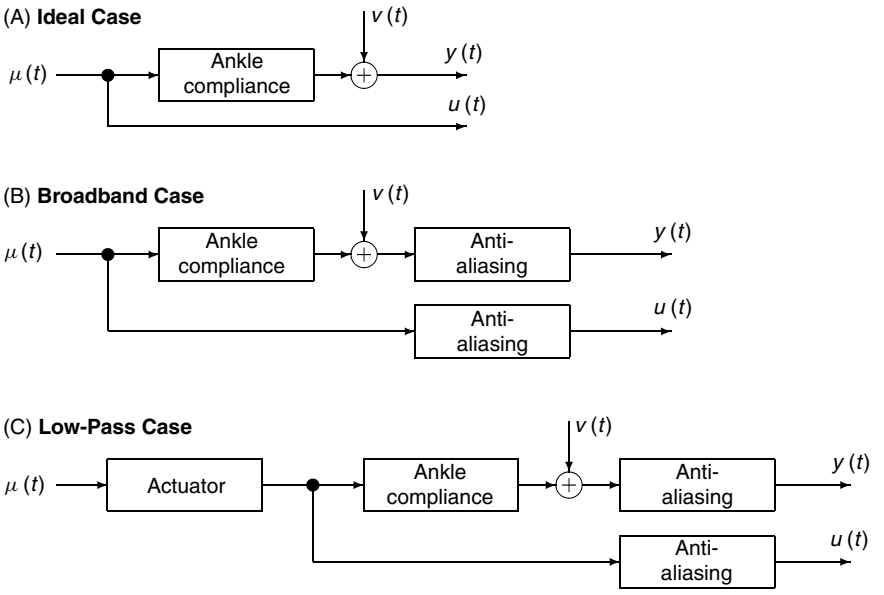

Figure 5.1 Block diagrams of the model configurations simulated. $\mu(t)$ is a white Gaussian noise signal, $u(t)$ and $y(t)$ are the measured input and output signals, and $v(t)$ is a white Gaussian noise signal. (A) The "ideal" case where the white noise input was applied directly to the system, and $u(t)$ and $y(t)$ were measured directly. (B) The "broadband" case with anti-aliasing filters added. (C) The "low-pass" case where $\mu(t)$ was low-pass filtered by actuator dynamics.

3. The "low-pass" case, Figure 5.1C, was generated by filtering a Gaussian white signal with a 50-Hz, third-order, low-pass Butterworth filter before applying it to the second-order system. This filter represents the dynamics of a mechanical actuator that will often modify inputs. Figures 5.2E and 5.2F illustrate a typical dataset for this case. Note that the input amplitude was scaled to have the same variance, or power, as the input for the "ideal" case.

In all three cases, 5000 points, corresponding to 10 s, of data were generated, and an independent, Gaussian white signal, $v(t)$, was added to the output of the "Ankle Compliance" block prior to anti-alias filtering and sampling. The noise amplitude was selected to give a signal-to-noise ratio (SNR) of 10 dB in each case.

Figures 5.3A and 5.3B show the power spectra for the input and the noise-free output signals for the three cases. The spectra were estimated using an averaged periodogram that broke the 5000 point records into 40 segments of 125 points.

The "ideal" and "broadband" spectra are virtually identical from 0 to 50 Hz. The "low-pass" spectrum was larger than, but almost proportional to, the ideal and broadband spectra from 0 to 40 Hz. Recall that the low-pass input was scaled to deliver the same total power to the system as the other inputs, so that it had more power at lower frequencies than the other two inputs. The "low-pass" input begins to roll off at 40 Hz due to the actuator dynamics while the "broadband" input spectrum starts to roll off at about 100 Hz due to the effects of the anti-aliasing filters.

Figure 5.3C shows the output spectra after the addition of 10 dB of Gaussian white output noise, before the anti-aliasing filters. It is evident from a comparison of these

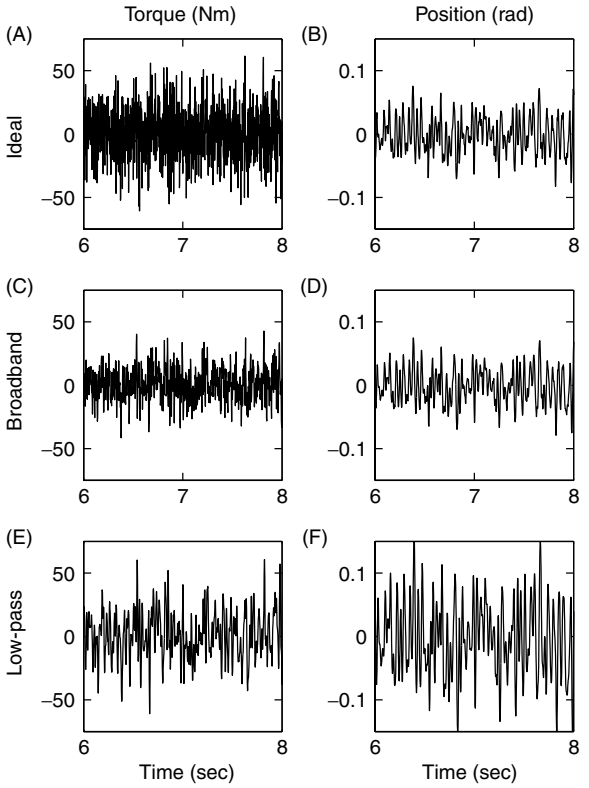

Figure 5.2 Simulated torque (input) and position (output) records from the ankle compliance model. (A) Ideal input torque. (B) Position output. (C, D) Broadband input and resulting output. (E, F) Low-pass input and resulting output. The system's impulse response is shown in Figure 5.4A.

spectra to those in Figure 5.3B that noise effects dominate at higher frequencies (>75 Hz) as the noise-free output is attenuated.

5.1.2 Model Evaluation

It will frequently be useful to measure how well a model describes an unknown system and to compare the accuracy of different models of the same system. Throughout this text, the *percent variance accounted for*, %VAF, will be used for this. The %VAF is defined as (Kearney and Hunter, 1983)

$$\%\text{VAF} = 100 \times \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \tag{5.1}$$

where y is a known vector, $\hat{y}$ is an estimate of it, and $\text{var}(x)$ is an estimate of the variance of the random variable x, given by

$$\text{var}(x) = \frac{1}{N}\sum_{t=1}^{N} x^2(t) - \left(\frac{1}{N}\sum_{t=1}^{N} x(t)\right)^2$$

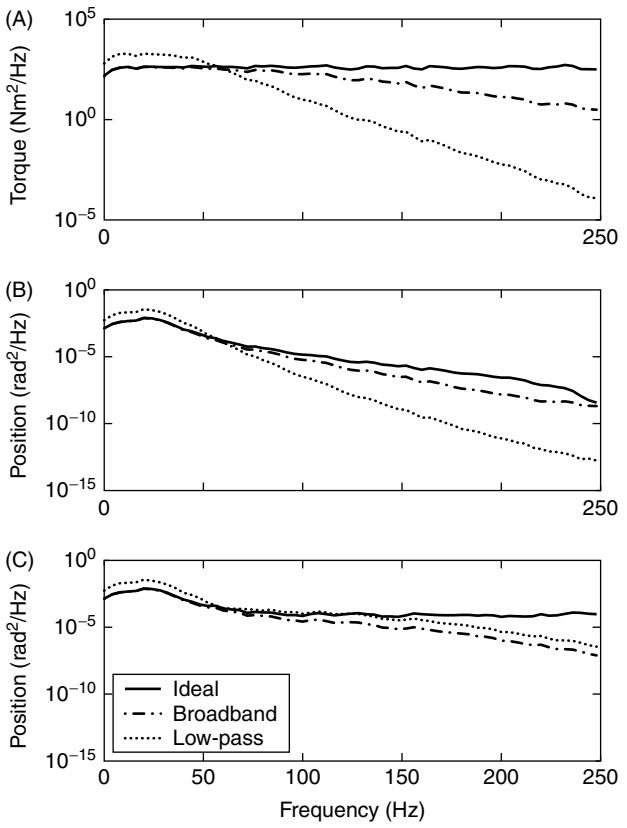

Figure 5.3 Input and output power spectra for the three simulation cases. (A) Input power spectra. (B) Spectra of the noise-free outputs. (C) Output spectra after the addition of 10 dB of white Gaussian noise.

A variety of other statistics have been used in the literature. Marmarelis and Marmarelis (1978) use the "normalized mean-square error" (NMSE), which is related to the %VAF by

$$\text{NMSE} = 1 - \frac{\%\text{VAF}}{100}$$

Ljung (1999) uses the "loss function," which is the sum of squared errors (SSE), to evaluate model accuracy. The SSE is also commonly used in the parameter estimation literature (Beck and Arnold, 1977). Measures based on the SSE have the disadvantage of depending explicitly on the length and amplitude of the input signal.

In simulation studies, the "true" (noise-free) output of the system is available. Consequently, a model's estimates can be evaluated by computing the %VAF between its output and the "true" output. Other descriptions of the true and estimated behavior (e.g., step responses, impulse response functions, frequency responses) may be compared similarly. Such statistics will be termed *model %VAFs*.

The situation is different for experimental data since the "true" system is unknown and the measured output will usually be corrupted by noise. Therefore, a model's accuracy

must be evaluated by comparing its output with that observed experimentally. Such %VAFs will be termed *prediction %VAFs*. These %VAFs may be computed from an *in sample prediction* made with the same data used to identify the model. However, this runs the risk of overestimating the model's predictive power since, if the model is overparameterized, it may describe the noise as well as the system dynamics. This can be avoided by using a *cross-validation* prediction where the %VAF is computed using a dataset that is different from that used for identification. This eliminates contributions from modeling noise and thus provides a more realistic estimate of the model accuracy.

The choice between in-sample and cross-validation prediction will depend upon the nature of the application and the amount of data available. Cross-validation is most appropriate when the model structure is not well-defined and extensive data are available. Conversely, in-sample prediction is appropriate if the objective is to estimate the parameters of a particular model structure as accurately as possible, or if there are limited data. In such cases, in-sample predictions are desirable since all available data are used in the identification and model estimates will improve with the length of the input data.

5.2 NONPARAMETRIC TIME DOMAIN MODELS

Chapter 3 described two nonparametric, time domain models: the impulse and step responses. This section will consider how to estimate these models from experimental data. Techniques to be discussed include (a) direct measurement, least-squares regression applied to input–output data and (b) cross-correlation between the input and the output.

5.2.1 Direct Estimation

The most straightforward way to determine a system's impulse response function (IRF) would be to apply an impulse to the input and record the response. Unfortunately, there are practical problems with this. First, it is not physically possible to generate an impulse; the best that can be achieved is a brief pulse of finite amplitude. Thus, even for a perfectly linear system, the response will be the convolution of the test pulse with the system's IRF rather than the IRF itself. If it is possible to apply a pulse that is much shorter than the IRF, then the resulting convolution may approximate the IRF very closely. This approach is feasible when the input can be controlled with high bandwidth (e.g., electrical inputs for neural systems; sound inputs for the auditory system; light inputs for the visual system). However, in many situations, physical constraints (e.g., inertia in biomechanical systems) limit the input bandwidth, so that the width of the test pulse will be comparable to or longer than that of the IRF.

In theory, the pulse shape can be deconvolved from the pulse response. For example, MATLAB has a function, `deconv`, which does this. In practice, however, deconvolution will amplify high-frequency noise so that IRF estimates become noisy. Thus, deconvolution of pulse responses is rarely practical.

Second, the noise level of direct IRF estimates will be determined by the ratio of the power in the input to that of the noise. Pulse inputs apply very little power because they are so short. Consequently, IRFs estimated directly will have poor SNRs unless the experimental data are very "clean."

In principle, the SNR of the IRF estimate could be improved by increasing the pulse amplitude. In practice, amplitudes large enough to obtain good SNRs will often

drive systems beyond their linear range (see Chapter 4). Alternatively, the SNR can be increased by averaging the responses to many pulses. This increases the experimental time required since the response to each pulse must be allowed to decay completely before applying the next one.

The input power could also be increased by applying a step input and differentiating the response to obtain the IRF. However, the power in step inputs is concentrated at low frequencies, and consequently the SNR will decrease as frequency increases. This problem will be compounded by the amplification of high-frequency noise associated with differentiation. As a result, this approach is generally only feasible for systems with low-pass dynamics and little noise.

5.2.2 Least-Squares Regression

Another approach to estimating the impulse response function is to rearrange the convolution sum (3.9), to give the IRF in terms of the input and output signals. If, as is usually the case, there is noise present, there will be no unique solution and it will be necessary to find the "best" solution—usually defined as that which predicts the output with the minimum mean-square error (MMSE). By choosing a "rich" input, as described below, it is possible to obtain good estimates of the IRF, $\hat{h}(\tau)$, even when the noise level is high.

To achieve this, first rewrite the discrete convolution, (3.9), as the matrix equation

$$\mathbf{y} = \mathbf{U}\mathbf{h} \tag{5.2}$$

where $\mathbf{U}$ is a matrix containing delayed copies of the input

$$\mathbf{U} = \begin{bmatrix} u(1) & 0 & 0 & \dots & 0 \\ u(2) & u(1) & 0 & \dots & 0 \\ u(3) & u(2) & u(1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u(N) & u(N-1) & u(N-2) & \dots & u(N-T+1) \end{bmatrix} \tag{5.3}$$

$\mathbf{y}$ is an N element vector containing the output, $y(t)$, at times $1, 2, \dots, N$ and $\mathbf{h}$ is a T element vector containing the discrete impulse response for lags $0, 1, \dots, T-1$.

In realistic cases, $y(t)$ is not available and it is necessary to work with the signal $z(t)$, which contains additive noise,

$$z(t) = y(t) + v(t) \tag{5.4}$$

where $v(t)$ is assumed to be independent of the input, $u(t)$.

Consequently, estimation of the IRF can be formulated as finding the MMSE solution to

$$\mathbf{z} = \mathbf{U}\mathbf{h} + \mathbf{v} \tag{5.5}$$

This equation has the same form as the regression problem defined in equation (2.32). It is linear in the parameters, so the least-squares solution can be determined using the normal equations, (2.33), as

$$\hat{\mathbf{h}} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{z} \tag{5.6}$$

The number of computations required to solve equation (5.6) is proportional to NT^3 (Golub and Van Loan, 1989), while storing $\mathbf{U}$ requires NT elements. Thus, this approach becomes computationally demanding if either N or T is large. The next section will present a more efficient approach to IRF estimation.

5.2.3 Correlation-Based Methods

The input–output, cross-correlation of a linear system driven by a white input is

$$\begin{aligned}\phi_{uy}(\tau) &= E[u(t-\tau)y(t)] \\ &= E\left[u(t-\tau)\sum_{k=0}^{T-1} h(k)u(t-k)\right] \\ &= \sigma_u^2 h(\tau)\end{aligned}$$

Thus, the system's cross-correlation function will be equal to the IRF scaled by the input variance. This relation may be used to estimate the IRF directly when white inputs can be applied. Indeed, the "reverse correlation" approach (de Boer and Kuyper, 1968), used to study the auditory system, uses exactly this approach.

The cross-correlation function can also be used to estimate IRFs with nonwhite inputs because of the close relationship between equation (5.6) and correlation functions. Thus, the kth element of the $\mathbf{U}^T\mathbf{z}$ term,

$$\mathbf{U}(:,k)^T\mathbf{z} = \sum_{j=1}^{N-k+1} u(j)z(j+k-1) \tag{5.7}$$

is a biased estimate of $\phi_{uz}(k-1)$ multiplied by N [see equation (2.18)].

There is a similar relationship between the Hessian, $\mathbf{U}^T\mathbf{U}$, and the input autocorrelation. The (i, j)th element of the Hessian is the product of $\mathbf{U}(:, i)$ and $\mathbf{U}(:, j)$ and can be written as the sum

$$\mathbf{U}(:,i)^T\mathbf{U}(:,j) = \sum_{k=1}^{N-i+1} u(k)u(k+i-j) \tag{5.8}$$

for $i \geq j$.* From equation (2.18), the biased estimate of $\phi_{uu}(i-j)$ is

$$\hat{\phi}_{uu}(i-j) = \frac{1}{N}\sum_{k=1}^{N-i+j} u(k)u(k+i-j)$$

so that

$$\mathbf{U}(:,i)^T\mathbf{U}(:,j) = N\hat{\phi}_{uu}(i-j) - \sum_{k=N-i+2}^{N-i+j} u(k)u(k+i-j)$$

Thus, for $N \gg T$, the Hessian is approximately equal to N times a matrix of autocorrelation coefficients:

$$\mathbf{\Phi_{uu}}(i,j) = \phi_{uu}(i-j) \tag{5.9}$$

*The Hessian is symmetric.

Consequently, if the data record is long compared to the system memory, the solution to the normal equations may be written in terms of input autocorrelation and the input–output cross-correlation:

$$\hat{\mathbf{h}} = \boldsymbol{\Phi}_{\mathbf{uu}}^{-1}\boldsymbol{\phi}_{\mathbf{uz}} \tag{5.10}$$

The auto- and cross-correlations may be computed efficiently using algorithms based on the FFT (Bendat and Piersol, 1986; Oppenheim and Schafer, 1989; Press et al., 1992), as discussed in Section 2.3.5. Furthermore, the autocorrelation matrix, $\boldsymbol{\Phi}_{\mathbf{uu}}$, is symmetric and the elements along each diagonal are equal; matrices with this structure are known as symmetric Toeplitz matrices and can be inverted rapidly using Levinson's algorithm (Golub and Van Loan, 1989). Thus, computing the correlations and solving equation (5.10) is much faster than constructing the regression matrix, $\mathbf{U}$, and solving the regression directly. This approach was suggested by Hunter and Kearney (1983), using a derivation based on the definitions of the auto- and cross-correlation functions.

5.2.3.1 Example: Identification of Human Joint Compliance Figure 5.4 shows the IRF estimates obtained with the cross-correlation approach of equation (5.10). Figure 5.4A shows the IRF of the simulated model, which is identical to Figure 3.3. Figures 5.4B and 5.4C show the estimates obtained using the ideal and broadband datasets, respectively, with no output noise. These are virtually identical to the theoretical result of Figure 5.4A. Figure 5.4D shows that when output noise is added to the broadband data, the IRF estimate contains some high-frequency noise; the model VAF was 97.3%. Figure 5.4E shows the IRF estimated with the noise-free, low-pass data; estimation error is clearly visible between lags of 0.08 and 0.1 s. This model had a VAF of 96.7%. Figure 5.4F shows the effects of adding output noise to the low-pass dataset; substantial estimation error is apparent and the model VAF drops to 64%.

The IRFs estimated using the low-pass filtered inputs contain high-frequency noise. Comparing Figures 5.4D and 5.4F, it is evident that the amplitude of the estimation errors increases as the input bandwidth decreases. The error also increases with the output-noise power. The next section will discuss how these errors arise and how they can be reduced.

5.2.3.2 Performance Issues For long records, where $N \gg T$, solving equation (5.10) is equivalent to performing a linear regression. Thus, equation (5.6) will give the IRF estimate, and the results of Section 2.4.2 can be used to analyze its performance. Therefore, $\hat{\mathbf{h}}$ will be an unbiased estimate of the IRF provided that:

1. The model structure is correct; that is, the underlying system is linear and its memory length is less than or equal to that of the IRF estimate.
2. The output noise is additive, zero-mean, and statistically independent of $u(t)$.

If, in addition, the measurement noise is white, the covariance matrix for the IRF estimate will be given by equation (2.41),

$$E[(\mathbf{h} - \hat{\mathbf{h}})(\mathbf{h} - \hat{\mathbf{h}})^T] = \sigma_v^2(\mathbf{U}^T\mathbf{U})^{-1}$$

where σ_v^2 is the output noise variance. Thus, as with any least-squares regression, the variance of the estimate will depend on the inverse of the Hessian, $\mathbf{H} = \mathbf{U}^T\mathbf{U}$. This is a

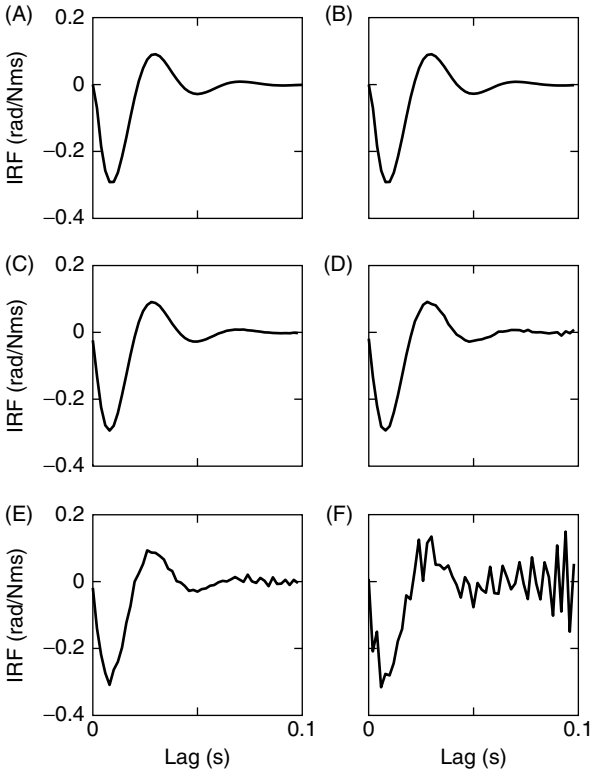

Figure 5.4 Least-squares estimates of the human ankle compliance IRF. (A) The theoretical IRF of the model. (B) IRF estimated from the "ideal" dataset. (C) IRF estimated from the "broadband" input and noise-free output. (D) IRF estimated from the "broadband" input and 10 dB of additive output noise. (E) IRF estimated with the "low-pass" input and noise-free output. (F) IRF estimated with "low-pass" input and 10 dB of additive output noise.

positive definite matrix and therefore will have the singular value decomposition (SVD),

$$\mathbf{H} = \mathbf{V}\mathbf{S}\mathbf{V}^T \tag{5.11}$$

where $\mathbf{S}$ is a real, non-negative diagonal matrix, and $\mathbf{V}$ is an orthogonal matrix (Golub and Van Loan, 1989). Consequently, its inverse can be written as

$$\mathbf{H}^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{V}^T \tag{5.12}$$

where $\mathbf{S}^{-1} = \text{diag}[1/s_1, 1/s_2, \ldots, 1/s_n]$.

Remember that $\mathbf{z} = \mathbf{y} + \mathbf{v}$ and $\mathbf{y} = \mathbf{U}\mathbf{h}$, so

$$\begin{aligned}\mathbf{U}^T\mathbf{z} &= \mathbf{U}^T\mathbf{U}\mathbf{h} + \mathbf{U}^T\mathbf{v} \\ &= \mathbf{V}\mathbf{S}\mathbf{V}^T\mathbf{h} + \mathbf{U}^T\mathbf{v}\end{aligned}$$

where the Hessian has been replaced by its SVD. Let $\boldsymbol{\zeta} = \mathbf{V}^T\mathbf{h}$ and $\boldsymbol{\eta} = \mathbf{V}^T(\mathbf{U}^T\mathbf{v})$ be the projections of the IRF, $\mathbf{h}$, and the estimated input-noise cross-correlation, $\mathbf{U}^T\mathbf{v}$,

respectively, onto the Hessian's singular vectors, $\mathbf{V}$. Then, $\mathbf{U}^T\mathbf{z}$ can be written as

$$\mathbf{U}^T\mathbf{z} = \mathbf{VS}\boldsymbol{\zeta} + \mathbf{V}\boldsymbol{\eta} \tag{5.13}$$

Substituting equations (5.12) and (5.13) into the IRF estimate, equation (5.6) yields

$$\begin{aligned}\hat{\mathbf{h}} &= \mathbf{VS}^{-1}\mathbf{V}^T(\mathbf{VS}\boldsymbol{\zeta} + \mathbf{V}\boldsymbol{\eta}) \\ &= \mathbf{V}\boldsymbol{\zeta} + \mathbf{VS}^{-1}\boldsymbol{\eta} \\ &= \sum_{i=1}^{T}\left(\zeta_i + \frac{\eta_i}{s_i}\right)\mathbf{v_i}\end{aligned} \tag{5.14}$$

where ζ_i is the ith element of the vector $\boldsymbol{\zeta}$, and $\mathbf{v_i} = \mathbf{V}(:, i)$ is the ith singular vector of $\mathbf{H}$.

It is evident from equation (5.14) that each term in $\hat{\mathbf{h}}$ contains two components: one due to the system, ζ_i, and the other due to the noise, η_i/s_i. Noise terms associated with small singular values, s_i, will be "amplified" by $1/s_i$. The smaller the singular value, the greater the amplification. The ratio of the largest and smallest singular values, the condition number (Golub and Van Loan, 1989), provides an upper bound on the relative level of noise amplification introduced by the matrix inversion.

Figure 5.5 shows the Hessians computed for the three simulation cases. For the white input, the autocorrelation is an impulse at zero lag, and the Hessian, shown in Figure 5.5A, is a diagonal matrix. For nonwhite inputs the autocorrelation becomes broader; the off-diagonal elements increase in magnitude as the bandwidth of the input changes from broadband to low-pass, as shown in Figures 5.5B and 5.5C, respectively.

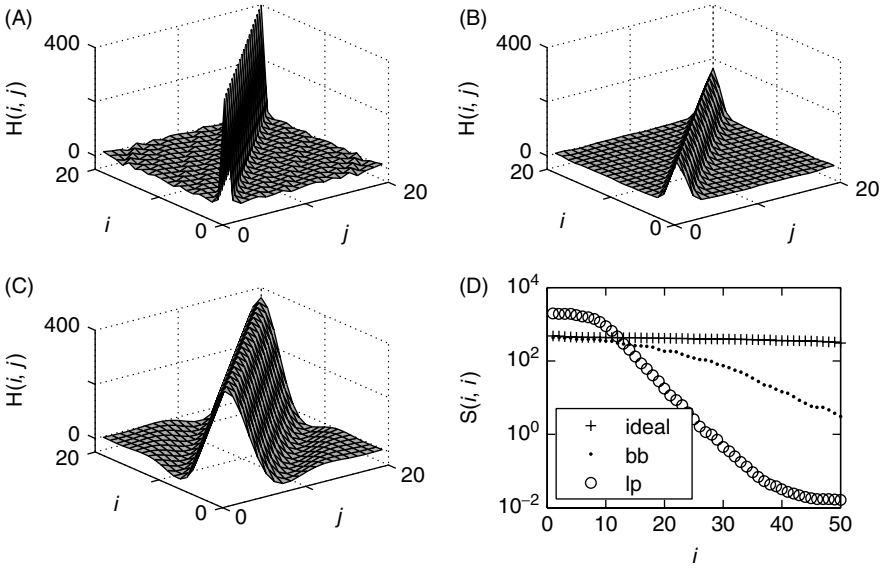

Figure 5.5 Hessians computed from the simulation inputs and their singular values. (A) Hessian for the "ideal" input, (B) "broadband" Hessian and (C) "low-pass" Hessian. (D) The singular values of the three Hessians shown in A, B, and C.

Figure 5.5D shows the singular values of these Hessians. For the ideal input, the singular values, shown as crosses, are almost constant, resulting in a condition number of 1.5. Inverting this Hessian should introduce no significant errors. In contrast, the singular values for the low-pass input, shown as open circles, vary over a wide range, resulting in a condition number of 1.2×10^5. Substantial errors can be expected to arise when inverting this Hessian.

5.2.3.3 Use of a Pseudo-Inverse Terms in equation (5.14) where $|\eta_i|/s_i > |\zeta_i|$ will add more "noise" than "signal" to the impulse response estimate, $\hat{h}$. Therefore, eliminating such components should improve the impulse response estimate. To do so, partition the SVD of $\hat{\mathbf{\Phi}}_{\mathbf{uu}}$ as follows:

$$\hat{\mathbf{\Phi}}_{\mathbf{uu}} = \begin{bmatrix}\mathbf{V_1} \; \mathbf{V_2}\end{bmatrix} \begin{bmatrix}\mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}\end{bmatrix} \begin{bmatrix}\mathbf{V_1}^T \\ \mathbf{V_2}^T\end{bmatrix} \tag{5.15}$$

where the subscript "1" refers to the terms to be retained, and "2" denotes terms to be dropped. Then, form a pseudo-inverse of $\hat{\mathbf{\Phi}}_{\mathbf{uu}}$:

$$\hat{\mathbf{\Phi}}_{\mathbf{uu}}^{\ddagger} = \mathbf{V_1}\mathbf{S_1}^{-1}\mathbf{V_1}^T \tag{5.16}$$

containing only terms that contribute significantly the output. The problem is to determine which terms are "significant," since eliminating terms from the pseudo-inverse will not only decrease the variance of the resulting estimate but will also introduce bias. The significance of each term's contribution can be evaluated using a cost function that incorporates a penalty term that increases with the number of model parameters. One such criterion is the minimum description length (MDL) cost function, defined by

$$\mathrm{MDL}(M) = \left(1 + \frac{M \log(N)}{N}\right) \sum_{t=1}^{N} (y(t) - \hat{y}(t, M))^2 \tag{5.17}$$

where M is the number of model parameters (i.e., singular vectors), and $\hat{y}(t, M)$ is the output of the M parameter model at time t. Thus, the sum of squared error is multiplied by the penalty term $(1 + M \log(N)/N)$, which increases monotonically with the number of parameters.

Building up a model term by term, along with testing each additional term for its statistical significance, would be arduous and time-consuming. However, this is not necessary since the contribution of each term to the output variance can be computed implicitly from the correlation functions as follows. The noise-free output variance of a linear system with a zero-mean input is

$$\frac{1}{N}\sum_{t=1}^{N} y^2(t) = \frac{1}{N}\sum_{t=1}^{N} \left(\sum_{\tau_1=0}^{T-1} h(\tau_1)u(t-\tau_1)\right)\left(\sum_{\tau_2=0}^{T-1} h(\tau_2)u(t-\tau_2)\right)$$

Rearranging the order of summations gives

$$\frac{1}{N}\sum_{t=1}^{N} y^2(t) = \sum_{\tau_1=0}^{T-1}\sum_{\tau_2=0}^{T-1} h(\tau_1)h(\tau_2)\left(\frac{1}{N}\sum_{t=1}^{N} u(t-\tau_1)u(t-\tau_2)\right)$$

The term in parentheses is $\mathbf{\Phi_{uu}}$, the Toeplitz structured autocorrelation matrix from equation (5.10). Rewriting the sums as vector multiplications gives

$$\frac{1}{N}\sum_{t=1}^{N} y^2(t) = \mathbf{h}^T \mathbf{\Phi_{uu}} \mathbf{h} \tag{5.18}$$

Thus the noise-free output variance may be computed rapidly from the IRF and input autocorrelation.

Expanding $\mathbf{\Phi_{uu}}$ in terms of mutually orthogonal singular vectors gives

$$\frac{1}{N}\sum_{t=1}^{N} y^2(t) = \mathbf{h}^T \mathbf{V}\mathbf{S}\mathbf{V}^T \mathbf{h} \tag{5.19}$$

$$= \sum_{i=1}^{T} s_i \left(\mathbf{v_i}^T \hat{\mathbf{h}}\right)^2 \tag{5.20}$$

Let γ_i be the output variance contributed by the ith singular value and its corresponding singular vector. Thus,

$$\gamma_i = s_i \left(\mathbf{v_i}^T \hat{\mathbf{h}}\right)^2 \tag{5.21}$$

Using $\boldsymbol{\gamma}$, the MDL corresponding to any set of model parameters may be computed implicitly from

$$\mathrm{MDL}(M) = \left(1 + \frac{M \log(N)}{N}\right)\left(\sigma_y^2 - \sum_{i=1}^{M} \gamma_i\right) \tag{5.22}$$

Similar relations can be derived for other criteria such as the Aikaike information criterion (Akaike, 1974).

5.2.3.4 Pseudo-Inverse Based Algorithm The complete algorithm for estimating the IRF of a linear system is then:

1. Estimate $\phi_{uu}(\tau)$ and $\phi_{uy}(\tau)$, using equation (2.18).
2. Compute an initial estimate of the IRF, $\hat{\mathbf{h}}$, using equation (5.10).
3. Compute the SVD of $\hat{\mathbf{\Phi}}_{\mathbf{uu}}$, and use equation (5.21) to calculate the variance that each singular vector contributes to the model output. Sort the result in decreasing order.
4. Calculate the MDL cost function using equation (5.22), for $M = 1, 2, \ldots, T$.
5. Choose the value of M that minimizes the MDL and retain only the M most significant terms for the final IRF estimate.

5.2.3.5 Example: Band-Limited Input Consider the identification of the human ankle compliance system using the noisy, low-pass dataset. Figure 5.5 showed that the singular values of the Hessian for this case will have a wide range so the matrix will be severely ill-conditioned. Equation (5.14) demonstrated that any noise projecting onto the smaller singular vectors will be amplified greatly as is clearly evident in Figure 5.4F.

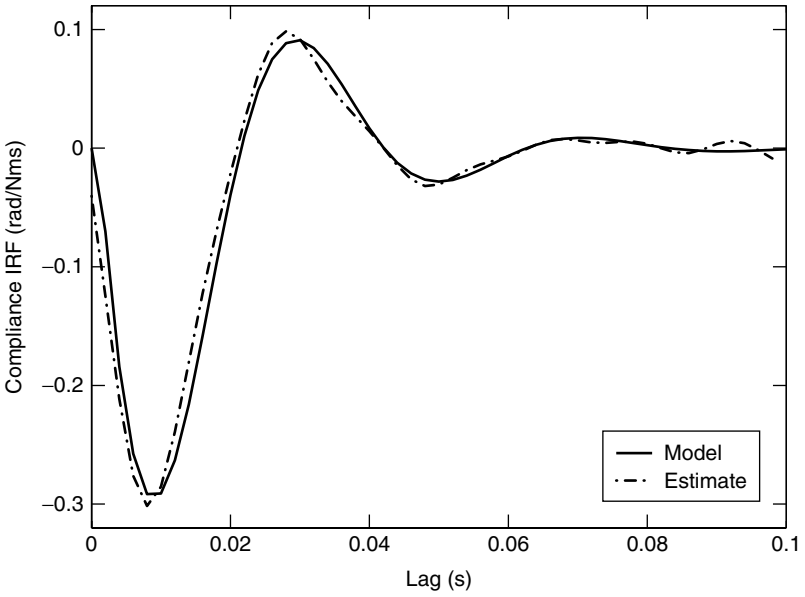

Figure 5.6 Compliance IRF estimated from the noise-corrupted, low-pass dataset using the pseudo-inverse algorithm. It has been superimposed on the IRF of the simulation model. The IRF estimated from the same data by equation (5.10) without the pseudo-inverse is shown in Figure 5.4F.

Figure 5.6 shows the IRF estimate obtained using the pseudo-inverse algorithm which retained only 16 of the 50 singular vectors. The close correspondence of the estimated (dashed line) and theoretical (solid line) IRFs demonstrates that the noise was virtually eliminated; the model VAF for this estimate was 97.3%, a substantial improvement over the 64% VAF obtained with the unmodified correlation method.

Bias is a concern with the pseudo-inverse algorithm; projecting the IRF estimate onto a reduced number of singular vectors may eliminate significant components of the IRF. The true IRF was known for this example so the bias error could be computed by projecting the true IRF onto the 34 discarded singular vectors. The bias error was only 0.13% VAF, insignificant compared to the total error of 2.69% VAF.

5.3 FREQUENCY RESPONSE ESTIMATION

Linear systems may also be modeled nonparametrically in the frequency domain by their frequency responses (see Section 3.2.2). This section describes several methods for estimating frequency response models.

5.3.1 Sinusoidal Frequency Response Testing

An obvious way to estimate the frequency response of a linear system is to apply a sine-wave input,

$$u(t) = A \sin(\omega t)$$

which, from equation (3.10), will generate the steady-state response:

$$y_{ss}(t) = A|H(j\omega)| \sin(\omega t + \Phi[H(j\omega)])$$

where $|H(j\omega)|$ denotes the magnitude of $H(j\omega)$ and $\Phi[H(j\omega)]$ its phase.

The system may be "identified" by applying a series of different sinusoids with different frequencies and measuring the steady state gain and phase. The resulting frequency response estimate, $\hat{H}(j\omega)$, can then be used with equation (3.12) to predict the response to any input. Frequency response curves are usually presented graphically as Bode diagrams where the gain (in decibels) and phase (in degrees) are shown as functions of logarithmic frequency. There are standard, graphical methods for extracting parametric frequency domain models (i.e., transfer functions) from Bode diagrams. [See Kamen (1990) for example, and M. Khoo's contribution (Khoo, 2000) to this series.]

5.3.2 Stochastic Frequency Response Testing

Equation (2.21) defined the spectrum of a signal as the Fourier transform of its autocorrelation,

$$S_{uu}(f) = \mathfrak{F}(\phi_{uu}(\tau))$$

Similarly, equation (2.24) defined the cross-spectrum as the Fourier transform of the cross-correlation,

$$S_{uy}(f) = \mathfrak{F}(\phi_{uy}(\tau))$$

Now, the cross-correlation between the input and output of a linear system is given by*

$$\begin{aligned}
\phi_{uy}(\tau) &= E[u(t)y(t+\tau)] \\
&= E\left[\int_0^\infty h(v)u(t)u(t+\tau-v)\,dv\right] \\
&= \int_0^\infty h(v)E[u(t)u(t+\tau-v)]\,dv \\
&= \int_0^\infty h(v)\phi_{uu}(\tau-v)\,dv
\end{aligned} \tag{5.23}$$

Fourier transforming (5.23) gives

$$S_{uy}(f) = H(f)S_{uu}(f) \tag{5.24}$$

Consequently, the frequency response may be estimated from the input power spectrum and the input–output cross-power spectrum.

$$\hat{H}(f) = \frac{\hat{S}_{uy}(f)}{\hat{S}_{uu}(f)}$$

Whatever method used to estimate it, $\hat{S}_{uy}(f)$ will be a complex number so the frequency response will have both magnitude (i.e., gain) and phase characteristics. Moreover, as shown in Section 2.3.3, the cross-correlation estimate is not biased by additive noise. Consequently, estimates of the cross-spectra will also be unbiased. However, if

*This can also be obtained by rearranging equation (5.10).

the noise level is high, long data records, and hence much averaging, may be required to reduce the random error to acceptable levels.

5.3.3 Coherence Functions

The coherence squared function is a real-valued function defined by

$$\gamma_{uy}^2(\omega) = \frac{|S_{uy}(\omega)|^2}{S_{uu}(\omega)S_{yy}(\omega)} \tag{5.25}$$

where $u(t)$ is the input and $y(t)$ is the output.

Consider the case where $y(t)$ is the output of a linear, time-invariant (LTI) system with IRF $h(\tau)$. The output autocorrelation will be

$$\begin{aligned}\phi_{yy}(\tau) &= E[y(t)y(t+\tau)] \\ &= \int_0^T \int_0^T h(\nu)h(\mu)E[u(t-\nu)u(t+\tau-\mu)]\,d\nu\,d\mu \\ &= \int_0^T \int_0^T h(\nu)h(\mu)\phi_{uu}(\tau-\nu+\mu)\,d\nu\,d\mu\end{aligned}$$

Fourier transforming both sides gives

$$S_{yy}(\omega) = |H(\omega)|^2 S_{uu}(\omega) \tag{5.26}$$

Thus, the coherence will be

$$\gamma_{uy}^2(\omega) = \frac{|H(\omega)S_{uu}(\omega)|^2}{S_{uu}(\omega)\left(|H(\omega)|^2 S_{uu}(\omega)\right)} \tag{5.27}$$

The auto-power spectrum is a real function, and thus it can be taken out of the squared magnitude in the numerator. This cancels the two $S_{uu}(\omega)$ factors in the denominator. Thus, the coherence between noiseless records of the input and output of a linear system will be unity at all frequencies where there is significant input power.

If the measured output contains additive noise [i.e., $z(t) = y(t) + v(t)$] that is uncorrelated with the input, then the output spectrum will be

$$S_{zz}(\omega) = S_{yy}(\omega) + S_{vv}(\omega)$$

and the coherence will be

$$\begin{aligned}\gamma_{uz}^2(\omega) &= \frac{|H(\omega)S_{uu}(\omega)|^2}{S_{uu}(\omega)\left(|H(\omega)|^2 S_{uu}(\omega) + S_{vv}(\omega)\right)} \\ &= \frac{1}{1+\left(\frac{S_{vv}(\omega)}{S_{yy}(\omega)}\right)}\end{aligned} \tag{5.28}$$

Thus the coherence squared can be interpreted as the fraction of the output variance due to the linear response to an input as a function of frequency.

Note that the extraneous signal, $v(t)$, need not necessarily be noise. It could result from an additional input uncorrelated with $u(t)$ or from nonlinearities in the system.

If $v(t)$ is the result of a nonlinearity in the system, it will be uncorrelated, but not statistically independent, from the input, $u(t)$. For example, the nonlinearity could be

represented by higher-order Wiener kernels, whose outputs would be orthogonal (i.e., uncorrelated) with that of the first-order kernel, the linearized model. However, the outputs of these higher-order nonlinear terms are functions of the input and are therefore not statistically independent of it.

Note also that in practice, the coherence must be computed from estimates of the auto- and cross-spectra. Thus, the accuracy of the estimated coherence function will be limited by the accuracy of the spectral estimates used to compute it.

5.3.3.1 Example: Joint Compliance in the Frequency Domain Figure 5.7 shows the results of frequency domain analyses of the three simulation cases from Figure 5.3. The magnitude and phase of the frequency responses are shown in Figures 5.7A and 5.7B with the squared coherence functions in Figure 5.7C. Results from the "ideal" data set are shown as solid lines; estimates from the noise-free, broadband data were almost identical and so are not shown. The gain curve is flat at low frequency, has a small resonant peak at 20 Hz, and then "rolls-off" at a constant rate as

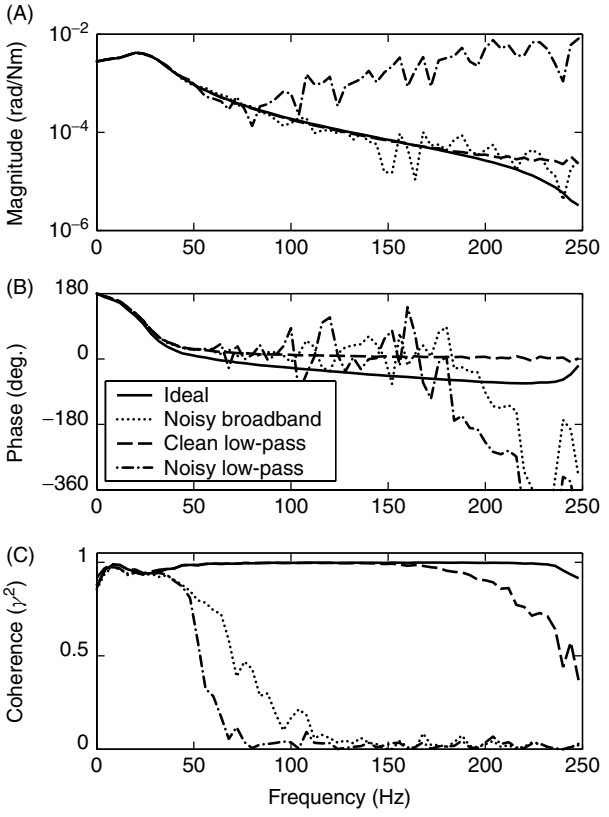

Figure 5.7 Frequency response and coherence estimates from the human ankle compliance simulations. The solid lines show the estimates obtained for the ideal case. Results for the noise-free, broadband case are indistinguishable from these and are not shown. The dotted lines show estimates for the noisy, broadband case. The dashed and dash–dotted lines show the results from the noise-free and noisy, low-pass datasets. (A) Frequency response magnitude. (B) Frequency response phase. (C) Coherence squared.

frequency continues to increase. The coherence estimate, shown in Figure 5.7C, is close to one over the entire frequency range.

The frequency response estimates for the noise-free, "low-pass" case are shown as dashed lines. Below 150 Hz they are identical to those from the ideal dataset. However, at higher frequencies, the "low-pass" frequency response plateaus, whereas the ideal frequency response continues to decrease. The coherence estimate is close to one at low frequency but then drops off, indicating that the frequency response estimate is becoming unreliable. These effects are due to the lower input power at high frequencies (see Figure 5.3A). As a result of the system's low-pass dynamics, the output power at these frequencies becomes much smaller so that numerical errors become significant, and the estimates become unreliable.

The results for the noisy, "broadband" case are shown as dotted lines. The frequency response for this dataset is virtually identical to the ideal estimate from DC to 40 Hz; small errors are evident between 40 and 100 Hz, and they increase at higher frequencies. The coherence is close to unity until 40 Hz, and then it begins to drop as the output power rolls off while the noise power remains constant; it is close to zero above 120 Hz.

Results for the noisy "low-pass" case are shown by the dash–dotted lines. The frequency response estimates match the "ideal" estimates between 0 and 40 Hz, but significant errors become evident at higher frequencies. Indeed, above 200 Hz, the estimated gain is larger than the low-frequency gain. However, the coherence for these estimates above 75 Hz is nearly zero, indicating that the model accounted for very little output power and thus is unreliable at these frequencies. At higher frequencies, the output signal was almost completely masked by the white, additive noise.

Several points become evident when comparing the time and frequency domain results. Both approaches yielded good results with the ideal and broadband data, with and without output noise. Similarly, both approaches produced acceptable results with the low-pass data, provided that there was no output noise, but both produced poor results with noisy data. The frequency-dependent nature of the estimation errors was demonstrated most clearly in the coherence plots. Since there is no direct analog to the coherence plot in the time domain, this insight can only be obtained using frequency domain methods.

5.4 PARAMETRIC METHODS

This book focuses on the identification of nonparametric models of nonlinear systems. However, the distinction between parametric and nonparametric methods is not absolute. Indeed, some of the methods for the identification of nonlinear, nonparametric models to be described in Chapter 8 are based on linear parametric methods. Consequently, a brief discussion of parametric techniques for linear identification is in order; more complete presentations are readily available elsewhere [e.g., Ljung (1999)].

5.4.1 Regression

The output error (OE) model, described by equation (3.19), can be expanded as follows:

$$\begin{aligned} z(k) = {} & b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m) \\ & - a_1 y(k-1) - \cdots - a_n y(k-n) + w(k) \end{aligned} \tag{5.29}$$

Form a regression matrix, $\mathbf{X}$, containing lagged inputs and (noise-free) outputs,

$$\mathbf{X}(k,:) = \begin{bmatrix} u(k) \ldots u(k-m)\ y(k-1) \ldots y(k-n) \end{bmatrix}$$

and then write equation (5.29) as

$$\mathbf{z} = \mathbf{X}\boldsymbol{\theta} + \mathbf{v} \tag{5.30}$$

An unbiased estimate of the parameters can be obtained from the least-squares solution,

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z} \tag{5.31}$$

provided that the noise $\mathbf{v}$ is not correlated with the regressors, $\mathbf{X}$.

In practice, however, $\mathbf{y}$ is not available, and the measured output, $\mathbf{z} = \mathbf{y} + \mathbf{v}$, must be used in the regression matrix. Even if $\mathbf{v}$ is white, the results will still be biased, since the Hessian will contain an extra diagonal term due to the noise in the lagged output measurements, because $\mathbf{v}^T\mathbf{v} \neq 0$.

5.4.2 Instrumental Variables

Estimation is more difficult if the output noise, $\mathbf{v}$, is not white. If lagged outputs are included in the regression matrix, $\mathbf{X}$, then $\mathbf{v}$ will be correlated with its columns and the resulting estimates will be biased. To obtain unbiased estimates, the correlation between the noise and the regression matrix must be eliminated. One approach is to project both regressors and the output onto a set of *instrumental variables*, which are correlated with the input signal but not the noise, before computing the regression.

To do this, select a matrix $\boldsymbol{\Psi}$ having the same dimension as the regressor, but whose columns are orthogonal to the noise, $\mathbf{v}$. Premultiply equation (5.30) by $\boldsymbol{\Psi}^T$, to give

$$\boldsymbol{\Psi}^T\mathbf{z} = \boldsymbol{\Psi}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\Psi}^T\mathbf{v}$$

Since $\boldsymbol{\Psi}^T\mathbf{v} = 0$, by construction, we have

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= (\boldsymbol{\Psi}^T\mathbf{X})^{-1}\boldsymbol{\Psi}^T\mathbf{z} \\ &= (\boldsymbol{\Psi}^T\mathbf{X})^{-1}\boldsymbol{\Psi}^T\mathbf{X}\boldsymbol{\theta} \end{aligned}$$

which can be solved exactly, provided that $\boldsymbol{\Psi}^T\mathbf{X}$ is nonsingular, to give an unbiased estimate of the parameter vector, $\boldsymbol{\theta}$.

The columns of $\boldsymbol{\Psi}$ are the *instrumental variables*, and they must be chosen correctly to obtain accurate results. Ideally, the instruments should be orthogonal to the noise, $\mathbf{v}$, but highly correlated with the regressors, so that $\boldsymbol{\Psi}^T\mathbf{X}$ is well-conditioned. One possibility is to use ordinary least-squares (5.31) to obtain a biased estimate, $[\hat{b}_0 \ldots \hat{b}_m \hat{a}_1 \ldots \hat{a}_n]$, of the model parameters. The input is then applied to this biased model to generate the sequence

$$x(t) = \sum_{i=0}^{m} b_i u(t-i) - \sum_{j=1}^{n} a_j x(t-j)$$

The instrumental variables are formed from $u(t)$ and $x(t)$ as follows:

$$\boldsymbol{\Psi}(t,:) = \begin{bmatrix} u(t) \ldots u(t-m)\ x(t-1) \ldots x(t-n) \end{bmatrix}$$

Since $v(t)$ is independent of the input, $u(t)$, and $x(t)$ is a function of $u(t)$, $x(t)$ and $v(t)$ are also independent of each other. Thus, the columns of $\mathbf{\Psi}$ will be orthogonal to the noise, $\mathbf{v}$, so $\mathbf{\Psi}$ can be used obtain an unbiased estimate of $\boldsymbol{\theta}$.

5.4.3 Nonlinear Optimization

Many model structures are not linear in their parameters, and therefore they cannot be identified using the normal equations. For example, consider the discrete-time state-space model, described in Section 3.4:

$$\mathbf{x(t+1)} = \mathbf{Ax(t)} + \mathbf{B}u(t)$$
$$y(t) = \mathbf{Cx(t)} + Du(t)$$

In Section 3.4, the state-space model was used to represent MIMO systems. For the sake of simplicity, this discussion is limited to SISO systems. Thus, $u(t)$, $y(t)$, and D are all scalars.

A closed-form expression for the output, $y(t)$, can be obtained by computing the IRF of the state-space model, given by equation (3.29), and substituting it into the discrete convolution, (3.9). Thus,

$$y(t) = Du(t) + \sum_{j=1}^{\infty} \mathbf{CA}^{j-1}\mathbf{B}u(t-j) \tag{5.32}$$

The output is a function of a power series in the **A** matrix rather than a linear function of the elements of **A**. Consequently, equation (5.32) cannot be rewritten as the multiplication of a regressor matrix and a parameter vector, and so the parameters cannot be estimated using the normal equations.

Iterative minimization can be used to estimate parameters for such models. These methods will be discussed extensively in Section 8.1, in conjunction with nonlinear model structures. A brief discussion follows.

First, it is necessary to postulate a model structure a priori and to make an initial estimate or guess, $\hat{\boldsymbol{\theta}}_\mathbf{0}$, for the parameter vector $\hat{\boldsymbol{\theta}}$. The final parameter estimates are then found using an iterative search.

Thus, if $\hat{\boldsymbol{\theta}}_\mathbf{k}$ represents the estimate after k updates, the next value will be of the form

$$\hat{\boldsymbol{\theta}}_{\mathbf{k+1}} = \hat{\boldsymbol{\theta}}_\mathbf{k} + \mathbf{d_k} \tag{5.33}$$

where $\mathbf{d_k}$ is the kth step. Generally, the goal is to choose the step sizes and directions such that the mean-square error,

$$V_N(\hat{\boldsymbol{\theta}}_\mathbf{k}) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(\hat{\boldsymbol{\theta}}_\mathbf{k}, t)\right)^2 \tag{5.34}$$

decreases with each step, that is, $V_N(\hat{\boldsymbol{\theta}}_{\mathbf{k+1}}) \leq V_N(\hat{\boldsymbol{\theta}}_\mathbf{k})$.

There are a variety of ways to select the step's direction and size. The simplest is to use the *gradient* of the error surface as the search direction and to use an adjustable step

size, μ_k:

$$\mathbf{d_k} = -\mu_k \left[\frac{\partial V_N(\boldsymbol{\theta})}{\partial \theta_1} \cdots \frac{\partial V_N(\boldsymbol{\theta})}{\partial \theta_M} \right]^T \Bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_\mathbf{k}} \tag{5.35}$$

Provided that the step sizes are small enough, the cost function will decrease, and the optimization will always converge to some minimum value. However, there is no guarantee that this minimum will be the global minimum; if the error surface is complex, the gradient search may get "caught" in a local minimum. Indeed, the algorithm may locate different minima for different initial parameter estimates.

Other gradient-based methods, such as the Newton, Gauss–Newton, and Levenberg–Marquardt techniques discussed at length in Section 8.1.3, use the curvature of the error surface to accelerate convergence (Press et al., 1992). All will eventually find a local minimum in the error surface but, as with the simple gradient technique, are not guaranteed to find the global minimum.

Stochastic search techniques, such as *simulated annealing* (Kirkpatric et al., 1983) and *genetic algorithms* (Holland, 1975), have been developed to avoid problems associated with local minima by searching broad regions of the parameter space at the cost of greatly increased convergence time. However, there is still no absolute guarantee that these will locate the global minimum in the error surface, and hence the optimal system model.

There is one further note of caution when using nonlinear minimization to identify a system. The model structure must be assumed a priori; if the model structure is incorrect, the "optimal" solution, even if it locates the global minimum for the model, will have little meaning and the parameter estimates may have little relation to those of the physical system.

5.5 NOTES AND REFERENCES

1. Ljung (1999) has been the standard reference for parametric linear system identification since the first edition appeared in 1987. Many of the algorithms described in Ljung (1999) are implemented in the MATLAB system identification toolbox.

5.6 COMPUTER EXERCISES

1. Run the file `ch5/mod1.m` to create a linear system called `model1`. Create an NLDAT object that contains a unit step, and filter it with `model1`. Differentiate the output, and compare it to the IRF of the model. Next, add noise to the step response, and differentiate the result. What happened?

2. Write an m-file to estimate an impulse response using explicit least-squares methods (i.e., generate the regressor matrix, **U**, and solve equation (5.6) directly). Generate a 500-point sequence of white Gaussian noise to use as the input. Compare the resulting Hessian, $\mathbf{U^T U}$, to the Toeplitz structured autocorrelation matrix obtained from

```
>>PHI=toeplitz(phixy(u,hlen));
```

Make a mesh plot of the difference between these two matrices. Where are the differences most pronounced? What happens to the difference if you double the data length?

3. Run the file `ch5/model3.m` to generate two signals, $u(t)$ and $y(t)$, from a linear system. Use spectral techniques to fit a system between $u(t)$ and $y(t)$, and evaluate the coherence. Now, fit a one-sided IRF between $u(t)$ and $y(t)$. What happened? How do you explain the apparent difference between the accuracies of the time and frequency domain models? What happens if you fit the time domain system between $y(t)$ and $u(t)$? What if you use a two-sided filter between $u(t)$ and $y(t)$?

4. Generate two sequences of white Gaussian noise of the same length; use one as a test input, $u(t)$, and use the other as measurement noise, $v(t)$. Compute the response of `model1`, the system from the first problem, to the test input, $u(t)$, and estimate the IRF of the system using `irf`. Add the noise sequence to the output, and repeat the identification. What is the SNR of the output? Scale the noise level so that the SNR is 10 dB, and repeat the identification. Use the first half of the data records for the identification. How does this affect the accuracy? Filter the input using a low-pass filter. How does the cutoff frequency of the prefilter affect the identification?