6.12 System Identification—The Easy Case

Assume that someone brings you a signal processing system enclosed in a black box. The box has two connectors, one marked *input* and the other *output*. Other than these labels there are no identifying marks or documentation, and nothing else is known about what is hidden inside. What can you learn about such a system? Is there some set of measurements and calculations that will enable you to accurately predict the system's output when an arbitrary input is applied? This task is known as *system identification*.

You can consider system identification as a kind of game between yourself and an opponent. The game is played in the following manner. Your opponent brings you the black box (which may have been specifically fabricated for the purpose of the game). You are given a specified finite amount of time to experiment with the system. Next your opponent specifies a test input and asks you for your prediction—were this signal to be applied what output would result? The test input is now applied and your prediction put to the test.

Since your opponent is an antagonist you can expect the test input to be totally unlike any input you have previously tried (after all, you don't have time to try *every possible* input). Your opponent may be trying to trick you in many ways. Is it possible to win this game?

This game has two levels of play. In this section we will learn how to play the easy version; in the next section we will make a first attempt at a strategy for the more difficult level. The easy case is when you are given complete control over the black box. You are allowed to apply controlled inputs and observe the resulting output. The difficult case is when you are not allowed to control the box at all. The box is already hooked up and operating. You are only allowed to observe the input and output.

The latter case is not only more difficult, it may not even be possible to pass the prediction test. For instance, you may be unlucky and during the entire time you observe the system the input may be zero. Or the input may contain only a single sinusoid and you are asked to predict the output when the input is a sinusoid of a different frequency. In such cases it is quite unreasonable to expect to be able to completely identify the hidden system. Indeed, this case is so much harder than the first that the term *system identification* is often reserved for it.

However, even the easy case is far from trivial in general. To see this consider a system that is not time-invariant. Your opponent knows that precisely at noon the system will shut down and its output will be zero thereafter. You are given until 11:59 to observe the system and give your prediction a few seconds before noon. Of course when the system is tested after noon your prediction turns out to be completely wrong! I think you will agree that the game is only fair if we limit ourselves to the identification of time-invariant systems.

Your opponent may still have a trick or two left! The system may have been built to be sensitive to a very specific trigger. For example, for almost every input signal the box may pass the signal unchanged; but for the trigger signal the output will be quite different! A signal that is different from the trigger signal in any way, even only having a slightly different amplitude or having an infinitesimal amount of additive noise, does not trigger the mechanism and is passed unchanged. You toil away trying a large variety of signals and your best prediction is that the system is simply an identity system. Then your opponent supplies the trigger as the test input and the system's output quite astounds you.

The only sensible way to avoid this kind of pitfall is to limit ourselves to linear systems. Linear systems may still be sensitive to specific signals. For example, think of a box that contains the identity system and in parallel a narrow band-pass filter with a strong amplifier. For most signals the output equals the input, but for signals in the band-pass filter's range the output is strongly amplified. However, for linear systems it is not possible to hide the trigger signal. Changing the amplitude or adding some noise will still allow triggering to occur, and once the effect is observed you may home in on it.

So the system identification game is really only fair for linear time-invariant systems, that is, for filters. It doesn't matter to us whether the filters are MA, AR, ARMA, or even without memory; that can be determined from your measurements. Of course since the black box is a real system, it is of necessity realizable as well, and in particular causal. Therefore from now on we will assume that the black box contains an unknown causal filter. If anyone offers to play the game without promising that the box contains a causal filter, don't accept the challenge!

Our task in this section is to develop a winning strategy for the easy case. Let's assume you are given one hour to examine the box in any way you wish (short of prying off the top). At the end of precisely one hour your opponent will reappear, present you with an input signal and ask you what you believe the box's response will be. The most straightforward way of proceeding would be to quickly apply as many different input signals as you can and to record the corresponding outputs. Then you win the game if your opponent's input signal turns out to be essentially one of the inputs you have checked. Unfortunately, there are very many possible inputs, and an hour is to short a time to test even a small fraction of them. To economize we can exploit the fact that the box contains a linear time-invariant system. If we have already tried input x_n there is no point in trying ax_n or x_{n-m}, but this still leaves a tremendous number of signals to check.

Our job can be made more manageable in two different ways, one of which relies on the time domain description of the input signal, and the other on its frequency domain representation. The frequency domain approach is based on Fourier's theorem that every signal can be written as the weighted sum (or integral) of basic sinusoids. Assume that you apply to the unknown system not every possible signal, but only every possible sinusoid. You store the system's response to each of these and wait for your opponent to appear. When presented with the test input you can simply break it down to its Fourier components, and exploit the filter's linearity to add the stored system responses with the appropriate Fourier coefficients.

Now this task of recording the system outputs is not as hard as it appears, since sinusoids are eigensignals of filters. When a sinusoid is input to a filter the output is a single sinusoid of the same frequency, only the amplitude and phase may be different. So you need only record these amplitudes and phases and use them to predict the system output for the test signal. For example, suppose the test signal turns out to be the sum of three sinusoids

$$x_n = X_1 \sin(\omega_1 n) + X_2 \sin(\omega_2 n) + X_3 \sin(\omega_3 n)$$

the responses of which had been measured to be

$$H_1 \sin(\omega_1 n + \phi_1), \qquad H_2 \sin(\omega_2 n + \phi_2), \qquad \text{and} \qquad H_3 \sin(\omega_3 n + \phi_3)$$

respectively. Then, since the filter is linear, the output is the sum of the three responses, with the Fourier coefficients.

$$y_n = H_1 X_1 \sin(\omega_1 n + \phi_1) + H_2 X_2 \sin(\omega_2 n + \phi_2) + H_3 X_3 \sin(\omega_3 n + \phi_3)$$

More generally, any finite duration or periodic test digital signal can be broken down by the DFT into the sum of a denumerable number of complex exponentials

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\mathrm{i} \frac{2\pi k}{N} n}$$

and the response of the system to each complex exponential is the same complex exponential multiplied by a number H_k.

$$H_k e^{\mathrm{i} \frac{2\pi k}{N} n}$$

Using these H_k we can predict the response to the test signal.

$$y_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k H_k e^{\mathrm{i}\frac{2\pi k}{N} n}$$

The H_k are in general complex (representing the gains and phase shifts) and are precisely the elements of the frequency response. A similar decomposition solves the problem for nonperiodic analog signals, only now we have to test a nondenumerable set of sinusoids.

The above discussion proves that the frequency response provides a complete description of a filter. Given the entire frequency response (i.e., the response of the system to all sinusoids), we can always win the game of predicting the response for an arbitrary input.

The frequency response is obviously a frequency domain quantity; the duality of time and frequency domains leads us to believe that there should be a complete description in the time domain as well. There is, and we previously called it the *impulse response*. To measure it we excite the system with a unit impulse (a Dirac delta function $\delta(t)$ for analog systems or a unit impulse signal $\delta_{n,0}$ for digital systems) and measure the output as a function of time (see equation 6.22). For systems without memory there will only be output for time $t = 0$, but in general the output will be nonzero over an entire time interval. A causal system will have its impulse response zero for times $t < 0$ but nonzero for $t \geq 0$. A system that is time-variant (and hence not a filter) requires measuring the response to all the SUIs, a quantity known as the Green's function.

Like the frequency response, the impulse response may be used to predict the output of a filter when an arbitrary input is applied. The strategy is similar to that we developed above, only this time we break down the test signal in the basis of SUIs (equation (2.26)) rather than using the Fourier expansion. We need only record the system's response to each SUI, expand the input signal in SUIs, and exploit the linearity of the system (as we have already done in Section 6.5). Unfortunately, the SUIs are not generally eigensignals of filters, and so the system's outputs will not be SUIs, and we need to record the entire output. However, unlike the frequency response where we needed to observe the system's output for an infinite number of basis functions, here we can capitalize on the fact that all SUIs are related by time shifts. Exploiting the time-invariance property of filters we realize that after measuring the response of an unknown system to a single SUI (e.g., the unit impulse at time zero), we may immediately deduce its response to all SUIs! Hence we need only apply a single input and record a single response in order to be able to predict the output of a filter when an arbitrary input is applied! The set of signals we must test in order to be able to predict the output of the system to an arbitrary input has been reduced to a single signal! This is the strength of the impulse response.

The impulse response may be nonzero only over a finite interval of time but exactly zero for all times outside this interval. In this case we say the system has a *finite impulse response*, or more commonly we simply call it an FIR filter. The MA systems studied in Sections 6.6 and 6.7 are FIR filters. To see this consider the noncausal three-point averaging system of equation (6.33).

$$y_n = \tfrac{1}{4}x_{n-1} + \tfrac{1}{2}x_n + \tfrac{1}{4}x_{n+1}$$

As time advances so does this window of time, always staying centered on the present. What happens when the input is an impulse? At time $n = \pm 1$ we find a $\frac{1}{4}$ multiplying the nonzero signal value at the origin, returning $\frac{1}{4}$; of course, the $n = 0$ has maximum output $\frac{1}{2}$. At any other time the output will be zero simply because the window does not overlap any nonzero input signal values. The same is the case for any finite combination of input signal values. Thus all the systems that have the form of equation (6.13), which we previously called FIR filters, are indeed FIR.

Let's explicitly calculate the impulse response for the most general causal moving average filter. Starting from equation (6.30) (but momentarily renaming the coefficients) and using the unit impulse as input yields

$$\begin{aligned} y_n &= \sum_{l=0}^{L} g_l \delta_{n-L+l,0} \\ &= g_0 \delta_{n-L,0} + g_1 \delta_{n-L+1,0} + g_2 \delta_{n-L+2,0} + \ldots + g_{L-1} \delta_{n-1,0} + g_L \delta_{n,0} \end{aligned}$$

which is nonzero only when $n = 0$ or $n = 1$ or $\ldots$ or $n = L$. Furthermore, when $n = 0$ the output is precisely $h_0 = g_L$, when $n = 1$ the output is precisely $h_1 = g_{L-1}$, etc., until $h_L = g_0$. Thus the impulse response of a general MA filter consists exactly of the coefficients that appear in the moving average sum, but in reverse order!

The impulse response is such an important attribute of a filter that it is conventional to reverse the definition of the moving average, and define the FIR filter via the *convolution* in which the indices run in opposite directions, as we did in equation (6.13).

It is evident that were we to calculate the impulse response of the nonterminating convolution of equation (6.14) it would consist of the coefficients as well; but in this case the impulse response would never quite become zero.

If we apply a unit impulse to a system and its output never dies down to zero, we say that the system is **I**nfinite **I**mpulse **R**esponse (IIR). Systems of the form (6.15), which we previously called IIR filters, can indeed sustain an impulse response that is nonzero for an infinite amount of time. To see this consider the simple case

$$y_n = x_n + \tfrac{1}{2} y_{n-1}$$

which is of the type of equation (6.15). For negative times n the output is zero, $y_n = 0$, but at time zero $y_0 = 1$, at time one $y_1 = \frac{1}{2}$ and thereafter y_n is halved every time. It is obvious that the output at time n is precisely $y_n = 2^{-n}$, which for large n is extremely small, but never zero.

Suppose we have been handed a black box and measure its impulse response. Although there may be many systems with this response to the unit impulse, there will be only one filter that matches, and the coefficients of equation (6.14) are precisely the impulse response in reverse order. This means that if we know that the box contains a filter, then measuring the impulse response is sufficient to uniquely define the system. In particular, we needn't measure the frequency response since it is mathematically derivable from the impulse response.

It is instructive to find this connection between the impulse response (the time domain description) and the frequency response (the frequency domain description) of a filter. The frequency response of the nonterminating convolution system

$$y_n = \sum_{i=-\infty}^{\infty} h_i x_{n-i}$$

is found by substituting a sinusoidal input for x_n, and for mathematical convenience we will use a complex sinusoid $x_n = e^{\mathrm{i}\omega n}$. We thus obtain

$$\begin{aligned} H(\omega)\, x_n = \quad y_n \quad &= \sum_{k=-\infty}^{\infty} h_k\, e^{\mathrm{i}\omega(n-k)} \\ &= \sum_{k=-\infty}^{\infty} h_k\, e^{-\mathrm{i}\omega k}\, e^{\mathrm{i}\omega n} \\ &= \quad H_k \qquad\quad x_n \end{aligned} \tag{6.53}$$

where we identified the Fourier transform of the impulse response h_k and the input signal. We have once again shown that when the convolution system has a sinusoidal input its output is the same sinusoid multiplied by a (frequency-dependent) gain. This gain is the frequency response, but here we have found the FT of the impulse response; hence the frequency response and the impulse response are an FT pair. Just as the time and frequency domain representations of signals are connected by the Fourier transform, the simplest representations of filters in the time and frequency domains are related by the FT.

EXERCISES

6.12.1 Find the impulse response for the following systems.

1. $y_n = x_n$
2. $y_n = x_n + x_{n-2} + x_{n-4}$
3. $y_n = x_n + 2x_{n-1} + 3x_{n-2}$
4. $y_n = \sum_i a_i x_{n-i}$
5. $y_n = x_n + y_{n-1}$
6. $y_n = x_n + \frac{1}{2}(y_{n-1} + y_{n-2})$

6.12.2 An ideal low-pass filter (i.e., one that passes without change signals under some frequency but entirely blocks those above it) is unrealizable. Prove this by arguing that the Fourier transform of a step function is nonzero over the entire axis and then invoking the connection between frequency response and impulse response.

6.12.3 When determining the frequency response we needn't apply each sinusoidal input separately; sinusoid orthogonality and filter linearity allow us to apply multiple sinusoids at the same time. This is what is done in probe signals (cf. exercise 2.6.4). Can we apply all possible sinusoids at the same time and reduce the number of input signals to one?

6.12.4 Since white noise contains all frequencies with the same amplitude, applying white noise to the system is somehow equivalent to applying all possible sinusoids. The *white noise response* is the response of a system to white noise. Prove that for linear systems the spectral amplitude of the white noise response is the amplitude of the frequency response. What about the phase delay portion of the frequency response?

6.12.5 The fact that the impulse and frequency responses are an FT pair derives from the general rule that the FT relates convolution and multiplication $\text{FT}(x * y) = \text{FT}(x)\text{FT}(y)$. Prove this general statement and relate it to the Wiener-Khintchine theorem.

6.12.6 Donald S. Perfectionist tries to measure the frequency response of a system by measuring the output power while injecting a slowly sweeping tone of constant amplitude. Unbeknownst to him the system contains a filter that passes most frequencies unattenuated, and amplifies a small band of frequencies. However, following the filter is a fast **A**utomatic **G**ain **C**ontrol (AGC) that causes all Donald's test outputs to have the same amplitude, thus completely masking the filter. What's wrong?

6.13 System Identification—The Hard Case

Returning to our system identification game, assume that your opponent presents you with a black box that is already connected to an input. We will assume first that the system is known to be an FIR filter of known length $L+1$. If the system is FIR of unknown length we need simply assume some extremely large $L+1$, find the coefficients, and discard all the zero coefficients above the true length.

The above assumption implies that the system's output at time n is

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_L x_{n-L}$$

and your job is to determine these coefficients a_l by simultaneously observing the system's input and output. It is clear that this game is riskier than the previous one. You may be very unlucky and during the entire time we observe it the system's input may be identically zero; or you may be very lucky and the input may be a unit impulse and we readily derive the impulse response.

Let's assume that the input signal was zero for some long time (and the output is consequently zero as well) and then suddenly it is turned on. We'll reset our clock to call the time of the first nonzero input time zero (i.e., x_n is identically zero for $n < 0$, but nonzero at $n = 0$). According to the defining equation the first output must be

$$y_0 = a_0 x_0$$

and since we observe both x_0 and y_0 we can easily find

$$a_0 = \frac{y_0}{x_0}$$

which is well defined since by definition $x_0 \neq 0$. Next, observing the input and output at time $n = 1$, we have

$$y_1 = a_0 x_1 + a_1 x_0$$

which can be solved

$$a_1 = \frac{y_1 - a_0 x_1}{x_0}$$

since everything needed is known, and once again $x_0 \neq 0$.

Continuing in this fashion we can express the coefficient a_n at time n in terms of $x_0 \ldots x_n$, $y_0 \ldots y_n$, and $a_0 \ldots a_{n-1}$, all of which are known. To see this explicitly write the equations

$$
\begin{aligned}
y_0 &= a_0 x_0 \\
y_1 &= a_0 x_1 + a_1 x_0 \\
y_2 &= a_0 x_2 + a_1 x_1 + a_2 x_0 \\
y_3 &= a_0 x_3 + a_1 x_2 + a_2 x_1 + a_3 x_0 \\
y_4 &= a_0 x_4 + a_1 x_3 + a_2 x_2 + a_3 x_1 + a_4 x_0
\end{aligned}
\tag{6.54}
$$

and so on, and note that these can be recursively solved

$$
\begin{aligned}
a_0 &= \frac{y_0}{x_0} \\
a_1 &= \frac{y_1 - a_0 x_1}{x_0} \\
a_2 &= \frac{y_2 - a_0 x_2 - a_1 x_1}{x_0} \\
a_3 &= \frac{y_3 - a_0 x_3 - a_1 x_2 - a_2 x_1}{x_0} \\
a_4 &= \frac{y_4 - a_0 x_4 - a_1 x_3 - a_2 x_2 - a_3 x_1}{x_0}
\end{aligned}
\tag{6.55}
$$

one coefficient at a time.

In order to simplify the arithmetic it is worthwhile to use linear algebra notation. We can write equation (6.54) in matrix form, with the desired coefficients on the right-hand side

$$
\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} x_0 & 0 & 0 & 0 & \dots \\ x_1 & x_0 & 0 & 0 & \dots \\ x_2 & x_1 & x_0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \end{pmatrix}
\tag{6.56}
$$

and identify the matrix containing the input values as being lower triangular and Toeplitz. The solution of (6.55) is simple due to the matrix being lower triangular. Finding the l^{th} coefficient requires l multiplications and subtractions and one division, so that finding all $L+1$ coefficients involves $\frac{1}{2}L(L+1)$ multiplications and subtractions and $L+1$ divisions.

The above solution to the 'hard' system identification problem was based on the assumption that the input signal was exactly zero for $n < 0$. What can we do in the common case when we start observing the signals at an arbitrary time before which the input was not zero? For notational simplicity let's assume that the system is known to be FIR with $L = 2$. Since we need to find three coefficients we will need three equations, so we observe three outputs, y_n, y_{n+1} and y_{n+2}. Now these outputs depend on five inputs, $x_{n-2}, x_{n-1}, x_n, x_{n+1}$, and x_{n+2} in the following way

$$\begin{aligned} y_n &= a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} \\ y_{n+1} &= a_0 x_{n+1} + a_1 x_n + a_2 x_{n-1} \\ y_{n+2} &= a_0 x_{n+2} + a_1 x_{n+1} + a_2 x_n \end{aligned} \tag{6.57}$$

which in matrix notation can be written

$$\begin{pmatrix} y_n \\ y_{n+1} \\ y_{n+2} \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & x_{n-2} \\ x_{n+1} & x_n & x_{n-1} \\ x_{n+2} & x_{n+1} & x_n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \tag{6.58}$$

or in other words $\underline{y} = \underline{\underline{X}}\,\underline{a}$, where $\underline{\underline{X}}$ is a nonsymmetric Toeplitz matrix. The solution is obviously $\underline{a} = \underline{\underline{X}}^{-1}\underline{y}$ but the three-by-three matrix is not lower triangular, and so its inversion is no longer trivial. For larger number of coefficients L we have to invert an $N = L + 1$ square matrix; although most direct N-by-N matrix inversion algorithms have computational complexity $O(N^3)$, it is possible to invert a general matrix in $O(N^{\log_2 7}) \sim O(N^{2.807})$ time. Exploiting the special characteristics of Toeplitz matrices reduces the computational load to $O(N^2)$.

What about AR filters?

$$y_n = x_n + \sum_{m=1}^{M} b_m y_{n-m}$$

Can we similarly find their coefficients in the hard system identification case? Once again, for notational simplicity we'll take $M = 3$. We have three unknown b coefficients, so we write down three equations,

$$\begin{aligned} y_n &= x_n + b_1 y_{n-1} + b_2 y_{n-2} + b_3 y_{n-3} \\ y_{n+1} &= x_{n+1} + b_1 y_n + b_2 y_{n-1} + b_3 y_{n-2} \\ y_{n+2} &= x_{n+2} + b_1 y_{n+1} + b_2 y_n + b_3 y_{n-1} \end{aligned} \tag{6.59}$$

or in matrix notation

$$\begin{pmatrix} y_n \\ y_{n+1} \\ y_{n+2} \end{pmatrix} = \begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix} + \begin{pmatrix} y_{n-1} & y_{n-2} & y_{n-3} \\ y_n & y_{n-1} & y_{n-2} \\ y_{n+1} & y_n & y_{n-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{6.60}$$

or simply $\underline{y} = \underline{x} + \underline{\underline{Y}}\,\underline{b}$. The answer this time is $\underline{b} = \underline{\underline{Y}}^{-1}(\underline{y} - \underline{x})$, which once again necessitates inverting a nonsymmetric Toeplitz matrix.

Finally, the full ARMA with $L = 2$ and $M = 3$

$$y_n = \sum_{l=0}^{L} a_l x_{n-l} + \sum_{m=1}^{M} b_m y_{n-m}$$

has six unknowns, and so we need to take six equations.

$$\begin{aligned}
y_n &= a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + b_1 y_{n-1} + b_2 y_{n-2} + b_3 y_{n-3} \\
y_{n+1} &= a_0 x_{n+1} + a_1 x_n + a_2 x_{n-1} + b_1 y_n + b_2 y_{n-1} + b_3 y_{n-2} \\
y_{n+2} &= a_0 x_{n+2} + a_1 x_{n+1} + a_2 x_n + b_1 y_{n+1} + b_2 y_n + b_3 y_{n-1} \\
y_{n+3} &= a_0 x_{n+3} + a_1 x_{n+2} + a_2 x_{n+1} + b_1 y_{n+2} + b_2 y_{n+1} + b_3 y_n \\
y_{n+4} &= a_0 x_{n+4} + a_1 x_{n+3} + a_2 x_{n+2} + b_1 y_{n+3} + b_2 y_{n+2} + b_3 y_{n+1} \\
y_{n+5} &= a_0 x_{n+5} + a_1 x_{n+4} + a_2 x_{n+3} + b_1 y_{n+4} + b_2 y_{n+3} + b_3 y_{n+2}
\end{aligned}$$

This can be written compactly

$$\begin{pmatrix} y_n \\ y_{n+1} \\ y_{n+2} \\ y_{n+3} \\ y_{n+4} \\ y_{n+5} \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & x_{n-2} & y_{n-1} & y_{n-2} & y_{n-3} \\ x_{n+1} & x_n & x_{n-1} & y_n & y_{n-1} & y_{n-2} \\ x_{n+2} & x_{n+1} & x_n & y_{n+1} & y_n & y_{n-1} \\ x_{n+3} & x_{n+2} & x_{n+1} & y_{n+2} & y_{n+1} & y_n \\ x_{n+4} & x_{n+3} & x_{n+2} & y_{n+3} & y_{n+2} & y_{n+1} \\ x_{n+5} & x_{n+4} & x_{n+3} & y_{n+4} & y_{n+3} & y_{n+2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{6.61}$$

and the solution requires inverting a six-by-six nonsymmetric non-Toeplitz matrix. The ARMA case is thus more computationally demanding than the pure MA or AR cases.

Up to now we have assumed that we observe x_n and y_n with no noise whatsoever. In all practical cases there will be at least some quantization noise, and most of the time there will be many other sources of additive noise. Due to this noise we will not get precisely the same answers when solving equations (6.58), (6.60), or (6.61) for two different times. One rather obvious tactic is to solve the equations many times and average the resulting coefficients. However, the matrix inversion would have to be performed a very large number of times and the equations (especially (6.60) and (6.61)) often turn out to be rather sensitive to noise. A much more successful tactic is to average *before* solving the equations, which has the advantages of providing more stable equations and requiring only a single matrix inversion.

Let's demonstrate how this is carried out for the MA case.

$$y_n = \sum_{k=0}^{L} a_k x_{n-k} \tag{6.62}$$

In order to average we multiply both sides by x_{n-q} and sum over as many n as we can get our hands on.

$$\sum_n y_n x_{n-q} = \sum_{k=0}^{L} a_k \sum_n x_{n-k} x_{n-q}$$

We define the x autocorrelation and the x-y crosscorrelation (see Chapter 9)

$$C_x(k) = \sum_n x_n x_{n-k} \qquad C_{yx}(k) = \sum_n y_n x_{n-k}$$

and note the following obvious symmetry.

$$C_x(-k) = C_x(k)$$

The deconvolution equations can now be written simply as

$$C_{yx}(q) = \sum_k a_k C_x(q-k) \tag{6.63}$$

and are called the Wiener-Hopf equations. For $L = 2$ the Wiener-Hopf equations look like this:

$$\begin{pmatrix} C_{yx}(0) \\ C_{yx}(1) \\ C_{yx}(2) \end{pmatrix} = \begin{pmatrix} C_x(0) & C_x(-1) & C_x(-2) \\ C_x(1) & C_x(0) & C_x(-1) \\ C_x(2) & C_x(1) & C_x(0) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

and from the aforementioned symmetry we immediately recognize the matrix as *symmetric* Toeplitz, a fact that makes them more stable and even faster to solve.

For a black box containing an AR filter, there is a special case where the input signal dies out (or perhaps the input happens to be an impulse). Once the input is zero

$$y_n = \sum_{m=1}^{M} b_m y_{n-m}$$

multiplying by y_{n-q} and summing over n we find

$$\sum_n y_n y_{n-q} = \sum_{m=1}^{M} b_m \sum_n y_{n-m} y_{n-q}$$

in which we identify y autocorrelations.

$$\sum_{m=1}^{M} C_y(|m-q|)b_m = C_y(q) \tag{6.64}$$

For $M = 3$ these equations look like this.

$$\begin{pmatrix} C_y(0) & C_y(1) & C_y(2) \\ C_y(1) & C_y(0) & C_y(1) \\ C_y(2) & C_y(1) & C_y(0) \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} C_y(1) \\ C_y(2) \\ C_y(3) \end{pmatrix}$$

These are the celebrated Yule-Walker equations, which will turn up again in Sections 9.8 and 9.9.

EXERCISES

6.13.1 Write a program that numerically solves equation (6.55) for the coefficients of a causal MA filter given arbitrary inputs and outputs. Pick such a filter and generate outputs for a pseudorandom input. Run your program for several different input sequences and compare the predicted coefficients with the true ones (e.g., calculate the squared difference). What happens if you try predicting with too long a filter? Too short a filter? If the input is a sinusoid instead of pseudorandom?

6.13.2 Repeat the previous exercise for AR filters (i.e., solve equation (6.60)). If the filter seems to be seriously wrong, try exciting it with a new pseudorandom input and comparing its output with the output of the intended system.

6.13.3 In the text we assumed that we knew the order L and M. How can we find the order of the system being identified?

6.13.4 Assume that y_n is related to x_n by a noncausal MA filter with coefficients $a_{-M} \dots a_M$. Derive equations for the coefficients in terms of the appropriate number of inputs and outputs.

6.13.5 In deriving the Wiener-Hopf equations we could have multiplied by y_{n-q} to get the equations

$$C_y(q) = \sum_k h_k C_{xy}(q-k)$$

rather than multiplying by x_{n-q}. Why didn't we?

6.13.6 In the derivation of the Wiener-Hopf equations we assumed that C_x and C_{yx} depend on k but not n. What assumption were we making about the noisy signals?