3

Parametric Methods for Rational Spectra

3.1 INTRODUCTION

The principal difference between the spectral-estimation methods of Chapter 2 and those in this chapter is that, in Chapter 2, we imposed no assumption on the studied signal (except stationarity). The *parametric* or *model-based methods* of spectral estimation assume that the signal satisfies a generating model with known functional form and then proceed by estimating the parameters in the assumed model. The signal's spectral characteristics of interest are then derived from the estimated model. In those cases where the assumed model is a close approximation to the reality, it is no wonder that the parametric methods provide more accurate spectral estimates than the nonparametric techniques. The nonparametric approach to PSD estimation remains useful, though, in applications where there is little or no information about the signal in question.

Our discussion of parametric methods for spectral estimation is divided into two parts. In this chapter, we discuss parametric methods for rational spectra, which form a dense set in the class of *continuous spectra* (see Section 3.2) [Anderson 1971; Wei 1990]; more precisely, we discuss methods for estimating the parameters in rational spectral models. The parametric methods of spectral analysis, unlike the nonparametric approaches, also require the selection of the *structure* (or order) of the spectral model. A review of methods that can be used to solve the structure-selection problem can be found in Appendix C. Furthermore, in Appendix B, we discuss the Cramér–Rao bound and the best accuracy achievable in the rational class of spectral models. However, we do not include detailed results on the statistical properties of the estimation methods discussed in the following sections, because (i) such results are readily available in the

literature [KAY 1988; PRIESTLEY 1981; SÖDERSTRÖM AND STOICA 1989]; (ii) parametric methods provide consistent spectral estimates and hence (for large sample sizes, at least) the issue of statistical behavior is not so critical; and (iii) a detailed statistical analysis is beyond the scope of an introductory course.

The second part of our discussion on parametric methods is contained in Chapter 4, where we consider *discrete spectra*, such as those associated with sinusoidal signals embedded in white noise. *Mixed spectra* (containing both continuous and discrete spectral components, such as in the case of sinusoidal signals corrupted by colored noise) are not covered explicitly in this text, but we remark that some methods in Chapter 4 can be extended to deal with such spectra as well.

3.2 SIGNALS WITH RATIONAL SPECTRA

A rational PSD is a rational function of $e^{-i\omega}$ (i.e., the ratio of two polynomials in $e^{-i\omega}$),

$$\phi(\omega) = \frac{\sum_{k=-m}^{m} \gamma_k e^{-i\omega k}}{\sum_{k=-n}^{n} \rho_k e^{-i\omega k}} \tag{3.2.1}$$

where $\gamma_{-k} = \gamma_k^*$ and $\rho_{-k} = \rho_k^*$. The Weierstrass theorem from calculus asserts that any continuous PSD can be approximated arbitrarily closely by a rational PSD of the form (3.2.1), provided the degrees m and n in (3.2.1) are chosen sufficiently large; that is, the rational PSDs form a *dense set in the class of all continuous spectra*. This observation partly motivates the significant interest in the model (3.2.1) for $\phi(\omega)$ among the researchers in the "spectral estimation community."

It is not difficult to show that, since $\phi(\omega) \geq 0$, the rational spectral density in (3.2.1) can be factored as

$$\boxed{\phi(\omega) = \left|\frac{B(\omega)}{A(\omega)}\right|^2 \sigma^2} \tag{3.2.2}$$

where σ^2 is a positive scalar and $A(\omega)$ and $B(\omega)$ are the polynomials:

$$\begin{aligned} A(\omega) &= 1 + a_1 e^{-i\omega} + \ldots + a_n e^{-in\omega} \\ B(\omega) &= 1 + b_1 e^{-i\omega} + \ldots + b_m e^{-im\omega} \end{aligned} \tag{3.2.3}$$

The result (3.2.2) can similarly be expressed in the Z-domain. With the notation $\phi(z) = \sum_{k=-m}^{m} \gamma_k z^{-k} / \sum_{k=-n}^{n} \rho_k z^{-k}$, we can factor $\phi(z)$ as

$$\phi(z) = \sigma^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)} \tag{3.2.4}$$

where, for example,

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$$

$$A^*(1/z^*) = [A(1/z^*)]^* = 1 + a_1^* z + \cdots + a_n^* z^n$$

Recall the notational convention in this text that we write, for example, $A(z)$ and $A(\omega)$ with the implicit understanding that, when we convert from a function of z to a function of ω, we use the substitution $z = e^{i\omega}$.

We note that the zeroes and poles of $\phi(z)$ are in symmetric pairs about the unit circle; if $z_i = re^{i\theta}$ is a zero (pole) of $\phi(z)$, then $(1/z_i^*) = (1/r)e^{i\theta}$ is also a zero (pole) (see Exercise 1.3). Under the assumption that $\phi(z)$ has no pole with modulus equal to one, the region of convergence of $\phi(z)$ includes the unit circle $z = e^{i\omega}$. The result that (3.2.1) can be written, as in (3.2.2) and (3.2.4), is called the *spectral factorization theorem*. (See, for example, [SÖDERSTRÖM AND STOICA 1989; KAY 1988].)

The next point of interest is to compare (3.2.2) with (1.4.9). This comparison leads to the following result:

The arbitrary rational PSD in (3.2.2) can be associated with a signal obtained by filtering white noise of power σ^2 through the rational filter with transfer function $H(\omega) = B(\omega)/A(\omega)$. (3.2.5)

The filtering referred to in (3.2.5) can be written in the time domain as

$$y(t) = \frac{B(z)}{A(z)} e(t) \tag{3.2.6}$$

or, alternatively, as

$$A(z)y(t) = B(z)e(t) \tag{3.2.7}$$

where $y(t)$ is the filter output and

z^{-1} = the unit delay operator ($z^{-k}y(t) = y(t-k)$)
$e(t)$ = white noise with variance σ^2

Hence, by means of the spectral factorization theorem, the parameterized model of $\phi(\omega)$ is turned into a model of the signal itself. The spectral estimation problem can then be reduced to a problem of *signal modeling*. In the following sections, we present several methods for estimating the parameters in the signal model (3.2.7) and in two of its special cases ($m = 0$, and $n = 0$).

A signal $y(t)$ satisfying the equation (3.2.6) is called an *autoregressive moving average* (ARMA or ARMA(n, m)) signal. If $B(z) = 1$ in (3.2.6) (i.e., $m = 0$ in (3.2.3)), then $y(t)$ is an *autoregressive* (AR or AR(n)) signal; and $y(t)$ is a *moving average* (MA or MA(m)) signal if $n = 0$. For easy reference, we summarize these naming conventions here:

$$\begin{aligned} &\text{ARMA}: && A(z)y(t) = B(z)e(t) \\ &\text{AR}: && A(z)y(t) = e(t) \\ &\text{MA}: && y(t) = B(z)e(t) \end{aligned} \tag{3.2.8}$$

By assumption, $\phi(\omega)$ is finite for all ω values; as a result, $A(z)$ cannot have any zero exactly on the unit circle. Furthermore, the poles and zeroes of $\phi(z)$ are in reciprocal pairs, so it is always possible to choose $A(z)$ to have all its zeroes strictly inside the unit disc. The corresponding model (3.2.6) is then said to be *stable*. If we assume, for simplicity, that $\phi(\omega)$ does not vanish at any ω, then—similarly to the preceding—we can choose the polynomial $B(z)$ so that it has all its zeroes inside the unit (open) disc. The corresponding model (3.2.6) is said to be of *minimum phase*. (See Exercise 3.1 for a motivation for the name "minimum phase.")

We remark that, in the previous paragraph, we actually provided a sketch of the proof of the spectral factorization theorem. That discussion also showed that the spectral factorization problem associated with a rational PSD has multiple solutions, with the stable and minimum-phase ARMA model being only one of them. In the next sections, we will consider the problem of estimating the parameters in this particular ARMA equation. When the final goal is the estimation of $\phi(\omega)$, focusing on the stable and minimum-phase ARMA model is no restriction.

3.3 COVARIANCE STRUCTURE OF ARMA PROCESSES

In this section, we derive an expression for the covariances of an ARMA process in terms of the parameters $\{a_i\}_{i=1}^n$, $\{b_i\}_{i=1}^m$, and σ^2. The expression provides a convenient method for estimating the ARMA parameters by replacing the true autocovariances with estimates obtained from data. Nearly all ARMA spectral estimation methods exploit this covariance structure either explicitly or implicitly; thus, it will be used widely in the remainder of the chapter.

Equation (3.2.7) can be written as

$$y(t) + \sum_{i=1}^{n} a_i y(t-i) = \sum_{j=0}^{m} b_j e(t-j), \qquad (b_0 = 1) \tag{3.3.1}$$

Multiplying (3.3.1) by $y^*(t-k)$ and taking expectation yields

$$r(k) + \sum_{i=1}^{n} a_i r(k-i) = \sum_{j=0}^{m} b_j E\left\{e(t-j)y^*(t-k)\right\} \tag{3.3.2}$$

Since the filter $H(z) = B(z)/A(z)$ is asymptotically stable and causal, we can write

$$H(z) = B(z)/A(z) = \sum_{k=0}^{\infty} h_k z^{-k}, \qquad (h_0 = 1)$$

which gives

$$y(t) = H(z)e(t) = \sum_{k=0}^{\infty} h_k e(t-k)$$

Then the term $E\{e(t-j)y^*(t-k)\}$ becomes

$$E\left\{e(t-j)y^*(t-k)\right\} = E\left\{e(t-j)\sum_{s=0}^{\infty} h_s^* e^*(t-k-s)\right\}$$

$$= \sigma^2 \sum_{s=0}^{\infty} h_s^* \delta_{j,k+s} = \sigma^2 h_{j-k}^*$$

where we use the convention that $h_k = 0$ for $k < 0$. Thus, equation (3.3.2) becomes

$$r(k) + \sum_{i=1}^{n} a_i r(k-i) = \sigma^2 \sum_{j=0}^{m} b_j h_{j-k}^* \tag{3.3.3}$$

In general, h_k is a nonlinear function of the $\{a_i\}$ and $\{b_i\}$ coefficients. However, $h_s = 0$ for $s < 0$, so equation (3.3.3) for $k \geq m+1$ reduces to

$$\boxed{r(k) + \sum_{i=1}^{n} a_i r(k-i) = 0, \qquad \text{for } k > m} \tag{3.3.4}$$

Equation (3.3.4) is the basis for many estimators of the AR coefficients of AR(MA) processes, as we will see.

3.4 AR SIGNALS

In the ARMA class, the *autoregressive* or *all-pole signals* constitute the type that is most frequently used in applications. The AR equation can model spectra with narrow peaks by placing zeroes of the A-polynomial in (3.2.2) (with $B(\omega) \equiv 1$) close to the unit circle. This is an important feature, because narrowband spectra are quite common in practice. In addition, the estimation of parameters in AR signal models is a well-established topic; the estimates are found by solving a system of linear equations, and the stability of the estimated AR polynomial can be guaranteed.

We consider two methods for AR spectral estimation. The first is based directly on the linear relationship between the covariances and the AR parameters derived in equation (3.3.4); it is called the Yule–Walker method. The second method is based on a least-squares solution of AR parameters using the time-domain equation $A(z)y(t) = e(t)$. This so-called "least-squares method" is closely related to the problem of linear prediction, as we shall see.

3.4.1 Yule–Walker Method

In this section, we focus on a technique for estimating the AR parameters that is called the *Yule–Walker (YW) method* [YULE 1927; WALKER 1931]. For AR signals, $m = 0$ and $B(z) = 1$. Thus, equation (3.3.4) holds for $k > 0$. Also, we have from equation (3.3.3) that

$$r(0) + \sum_{i=1}^{n} a_i r(-i) = \sigma^2 \sum_{j=0}^{0} b_j h_j^* = \sigma^2 \tag{3.4.1}$$

Combining (3.4.1) and (3.3.4) for $k = 1, \ldots, n$ gives the following system of linear equations:

$$\begin{bmatrix} r(0) & r(-1) & \ldots & r(-n) \\ r(1) & r(0) & & \vdots \\ \vdots & & \ddots & r(-1) \\ r(n) & \ldots & & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.4.2}$$

These equations are called the *Yule–Walker equations* or *normal equations*; they form the basis of many AR estimation methods. If $\{r(k)\}_{k=0}^{n}$ were known, we could solve (3.4.2) for

$$\theta = [a_1, \ldots, a_n]^T \tag{3.4.3}$$

by using all but the first row of (3.4.2)

$$\begin{bmatrix} r(1) \\ \vdots \\ r(n) \end{bmatrix} + \begin{bmatrix} r(0) & \cdots & r(-n+1) \\ \vdots & \ddots & \vdots \\ r(n-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.4.4}$$

or, with obvious definitions,

$$r_n + R_n \theta = 0 \tag{3.4.5}$$

The solution is $\theta = -R_n^{-1} r_n$. Once θ is found, σ^2 can be obtained from the first row of (3.4.2) or, equivalently, from (3.4.1).

The Yule–Walker method for AR spectral estimation is based directly on (3.4.2). Given data $\{y(t)\}_{t=1}^{N}$, we first obtain sample covariances $\{\hat{r}(k)\}_{k=0}^{n}$ by using the standard biased ACS estimator (2.2.4). We insert these ACS estimates in (3.4.2) and solve for $\hat{\theta}$ and $\hat{\sigma}^2$, as explained above in the known-covariance case.

Note that the covariance matrix in (3.4.2) can be shown to be positive definite for any n, and hence the solution to (3.4.2) is unique [Söderström and Stoica 1989]. When the covariances are replaced by standard biased ACS estimates, the matrix can be shown to be positive definite for any sample (not necessarily generated by an AR equation) that is not identically equal to zero; see the remark in the next section for a proof.

To explicitly stress the dependence of θ and σ^2 on the order n, we can write (3.4.2) as

$$\boxed{R_{n+1} \begin{bmatrix} 1 \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \end{bmatrix}} \tag{3.4.6}$$

We will return to this equation in Section 3.5.

3.4.2 Least-Squares Method

The Yule–Walker method for estimating the AR parameters is based on equation (3.4.2) with the true covariance elements $\{r(k)\}$ replaced by the sample covariances $\{\hat{r}(k)\}$. In this section, we derive another type of AR estimator, one based on a least-squares (LS) minimization criterion using the time-domain relation $A(z)y(t) = e(t)$. We develop the LS estimator by considering the closely related problem of *linear prediction*. We then interpret the LS method as a Yule–Walker-type method that uses a different estimate of R_{n+1} in equation (3.4.6).

We first relate the Yule–Walker equations to the linear prediction problem. Let $y(t)$ be an AR process of order n. Then $y(t)$ satisfies

$$\begin{aligned} e(t) &= y(t) + \sum_{i=1}^{n} a_i y(t-i) = y(t) + \varphi^T(t)\theta \\ &\triangleq y(t) + \hat{y}(t) \end{aligned} \tag{3.4.7}$$

where $\varphi(t) = [y(t-1), \ldots, y(t-n)]^T$. We interpret $\hat{y}(t)$ as a *linear prediction* of $y(t)$ from the n previous samples $y(t-1), \ldots, y(t-n)$, and we interpret $e(t)$ as the corresponding *prediction error*. See Complement 3.9.1 and also Exercises 3.3–3.5 for more discussion on this and other related linear prediction problems. The vector θ that minimizes the prediction error variance $\sigma_n^2 \triangleq E\left\{|e(t)|^2\right\}$ is the AR coefficient vector in (3.4.6), as we will show. From (3.4.7), we have

$$\begin{aligned} \sigma_n^2 = E\left\{|e(t)|^2\right\} &= E\left\{\left[y^*(t) + \theta^*\varphi^c(t)\right]\left[y(t) + \varphi^T(t)\theta\right]\right\} \\ &= r(0) + r_n^*\theta + \theta^* r_n + \theta^* R_n \theta \end{aligned} \tag{3.4.8}$$

where r_n and R_n are defined in equations (3.4.4)–(3.4.5). The vector θ that minimizes (3.4.8) is given (see Result R34 in Appendix A) by

$$\theta = -R_n^{-1} r_n \tag{3.4.9}$$

with corresponding minimum prediction error

$$\sigma_n^2 = r(0) - r_n^* R_n^{-1} r_n \tag{3.4.10}$$

Equations (3.4.9) and (3.4.10) are exactly the Yule–Walker equations in (3.4.5) and (3.4.1) (or, equivalently, in (3.4.6)). Thus, we see that the Yule–Walker equations can be interpreted as the solution to the problem of finding the best linear predictor of $y(t)$ from its n most recent past samples. For this reason, AR modeling is sometimes referred to as *linear predictive modeling*.

The least-squares AR estimation method is based on a finite-sample approximate solution of the above minimization problem. Given a finite set of measurements $\{y(t)\}_{t=1}^{N}$, we approximate

the minimization of $E\left\{|e(t)|^2\right\}$ by the finite-sample cost function

$$f(\theta) = \sum_{t=N_1}^{N_2} |e(t)|^2 = \sum_{t=N_1}^{N_2} \left| y(t) + \sum_{i=1}^{n} a_i y(t-i) \right|^2$$

$$= \left\| \begin{bmatrix} y(N_1) \\ y(N_1+1) \\ \vdots \\ y(N_2) \end{bmatrix} + \begin{bmatrix} y(N_1-1) & \cdots & y(N_1-n) \\ y(N_1) & \cdots & y(N_1+1-n) \\ \vdots & & \vdots \\ y(N_2-1) & \cdots & y(N_2-n) \end{bmatrix} \theta \right\|^2$$

$$\triangleq \|y + Y\theta\|^2 \tag{3.4.11}$$

where we assume $y(t) = 0$ for $t < 1$ and $t > N$. The vector θ that minimizes $f(\theta)$ is given (per Result R32 in Appendix A) by

$$\boxed{\hat{\theta} = -(Y^*Y)^{-1}(Y^*y)} \tag{3.4.12}$$

where, as seen from (3.4.11), the definitions of Y and y depend on the choice of (N_1, N_2) considered. If $N_1 = 1$ and $N_2 = N + n$, we have

$$y = \left[\begin{array}{c} y(1) \\ y(2) \\ \vdots \\ \hline y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right], \quad Y = \left[\begin{array}{cccc} 0 & 0 & \dots & 0 \\ y(1) & 0 & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \hline y(n) & y(n-1) & \cdots & y(1) \\ y(n+1) & y(n) & \cdots & y(2) \\ \vdots & & & \vdots \\ y(N-1) & y(N-2) & \cdots & y(N-n) \\ \hline y(N) & y(N-1) & \cdots & y(N-n+1) \\ 0 & y(N) & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & y(N) \end{array}\right] \tag{3.4.13}$$

Notice the Toeplitz structure of Y, and also notice that y matches this Toeplitz structure when it is appended to the left of Y; that is, $[y|Y]$ also shares this Toeplitz structure.

The two most common choices for N_1 and N_2 are the following:

- $N_1 = 1$, $N_2 = N + n$ (considered previously). This choice yields the so-called *autocorrelation method.*
- $N_1 = n + 1$, $N_2 = N$. This choice corresponds to removing the first n and last n rows of Y and y in equation (3.4.13) and, hence, eliminates all the arbitrary zero values there. The estimate (3.4.12) with this choice of (N_1, N_2) is often named the *covariance method.* We refer to this method as the *covariance LS method* or the *LS method*.

Other choices for N_1 and N_2 have also been suggested. For example, the *prewindow method* uses $N_1 = 1$ and $N_2 = N$, and the *postwindow method* uses $N_1 = n + 1$ and $N_2 = N$.

The least-squares methods can be interpreted as approximate solutions to the Yule–Walker equations in (3.4.4) by recognizing that Y^*Y and Y^*y are, to within a multiplicative constant, finite-sample estimates of R_n and r_n, respectively. In fact, it is easy to show that, for the autocorrelation method, the elements of $(Y^*Y)/N$ and $(Y^*y)/N$ are *exactly* the biased ACS estimates (2.2.4) used in the Yule–Walker AR estimate. Writing $\hat{\theta}$ in (3.4.12) as

$$\hat{\theta} = -\left[\frac{1}{N}(Y^*Y)\right]^{-1}\left[\frac{1}{N}(Y^*y)\right]$$

we see the following as a consequence:

> The autocorrelation method of least-squares AR estimation is equivalent to the Yule–Walker method.

Remark: We can now prove a claim made in the previous subsection: that the matrix Y^*Y in (3.4.12), with Y given by (3.4.13), is positive definite for any sample $\{y(t)\}_{t=1}^{N}$ that is not identically equal to zero. To prove this claim, it is necessary and sufficient to show that $\text{rank}(Y) = n$. If $y(1) \neq 0$, then clearly $\text{rank}(Y) = n$. If $y(1) = 0$ and $y(2) \neq 0$, then again we clearly have $\text{rank}(Y) = n$, and so on. ■

For the LS estimator, $(Y^*Y)/(N - n)$ and $(Y^*y)/(N - n)$ are unbiased estimates of R_n and r_n in equations (3.4.4) and (3.4.5), and they do not use any measurement data outside the available interval $1 \leq t \leq N$. On the other hand, the matrix $(Y^*Y)/(N - n)$ is not Toeplitz, so the Levinson–Durbin or Delsarte–Genin algorithms in the next section cannot be used (although similar fast algorithms for the LS method have been developed; see, for example, [Marple 1987]).

As N increases, the difference between the covariance matrix estimates used by the Yule–Walker and the LS methods diminishes. Consequently, for large samples (i.e., for $N \gg 1$), the YW and LS estimates of the AR parameters nearly coincide.

For small or medium sample lengths, the Yule–Walker and covariance LS methods might behave differently. *First*, the estimated AR model obtained via the Yule–Walker method is always guaranteed to be *stable* (see, for example, [Stoica and Nehorai 1987] and Exercise 3.8), whereas the estimated LS model could be unstable. For applications in which one is interested in the AR model (and not just the AR spectral estimate), stability of the model is often an important requirement. It may, therefore, be thought that the potential instability of the AR model provided by the LS method is a significant drawback of this method. However, estimated LS models that are unstable appear infrequently; moreover, when they do occur, there are simple means to "stabilize" them (for instance, by reflecting the unstable poles inside the unit circle). Hence, to conclude this point, the lack of guaranteed stability is a drawback of the LS method, when compared with the Yule–Walker method, but often not a serious one.

Second, the LS method has been found to be more accurate than the Yule–Walker method, in the sense that the estimated parameters of the former are on the average closer to the true values than those of the latter [Marple 1987; Kay 1988]. Because the finite-sample statistical analysis of these methods is underdeveloped, a theoretical explanation of this behavior is not possible at

this time. Only heuristic explanations are available. One such explanation is that the assumption that $y(t) = 0$ outside the interval $1 \le t \le N$, and the corresponding zero elements in Y and y, result in bias in the Yule–Walker estimates of the AR parameters. When N is not much greater than n, this bias can be significant.

3.5 ORDER-RECURSIVE SOLUTIONS TO THE YULE–WALKER EQUATIONS

In most applications, *a priori* information about the true order n is lacking, so AR models with different orders have to be tested. Hence, the Yule–Walker system of equations, (3.4.6), has to be solved for $n = 1$ up to $n = n_{\max}$ (some prespecified maximum order); see Appendix C. By using a general solving method, this task requires $\mathcal{O}(n_{\max}^4)$ flops. This can be a significant computational burden if $n_{\max}$ is large. This is, for example, the case in the applications dealing with narrowband signals, where values of 50 or even 100 for $n_{\max}$ are not uncommon. In such applications, it can be important to reduce the number of flops required to calculate $\{\theta_n, \sigma_n^2\}$ in (3.4.6). In order to be able to do so, the special algebraic structure of (3.4.6) should be exploited, as explained next.

The matrix R_{n+1} in the Yule–Walker system of equations is highly structured: it is *Hermitian* and *Toeplitz*. The first algorithm that exploited this fact to compute $\{\theta_n, \sigma_n^2\}_{n=1}^{n_{\max}}$ in $n_{\max}^2$ flops was the *Levinson–Durbin algorithm* (LDA) [Levinson 1947; Durbin 1960]. The number of flops required by the LDA is on the order of $n_{\max}$ times smaller than that required by a general linear-equation solver to compute $(\theta_{n_{\max}}, \sigma_{n_{\max}}^2)$, and on the order of $n_{\max}^2$ times smaller than that required by a general linear-equation solver to compute $\{\theta_n, \sigma_n^2\}_{n=1}^{n_{\max}}$. The LDA is discussed in Section 3.5.1. In Section 3.5.2, we present another algorithm, the *Delsarte–Genin algorithm* (DGA), also named the *split-Levinson algorithm*, which, in the case of real-valued signals, is about two times faster than the LDA [Delsarte and Genin 1986].

Both the LDA and DGA solve, recursively in the order n, equation (3.4.6). The only requirement is that the matrix there be positive definite, Hermitian, and Toeplitz. Thus, the algorithms apply equally well to the Yule–Walker AR estimator (or, equivalently, the autocorrelation least-squares AR method), in which the "true" ACS elements are replaced by estimates. Hence, to cover both cases simultaneously, in what follows,

$$\boxed{\rho_k \text{ is used to represent either } r(k) \text{ or } \hat{r}(k).} \tag{3.5.1}$$

By using the preceding convention, we have

$$R_{n+1} = \begin{bmatrix} \rho_0 & \rho_{-1} & \cdots & \rho_{-n} \\ \rho_1 & \rho_0 & & \vdots \\ \vdots & & \ddots & \rho_{-1} \\ \rho_n & \cdots & \rho_1 & \rho_0 \end{bmatrix} = \begin{bmatrix} \rho_0 & \rho_1^* & \cdots & \rho_n^* \\ \rho_1 & \rho_0 & & \vdots \\ \vdots & & \ddots & \rho_1^* \\ \rho_n & \cdots & \rho_1 & \rho_0 \end{bmatrix} \tag{3.5.2}$$

The following notational convention will also be used frequently in this section. For a vector $x = [x_1 \dots x_n]^T$, we define

$$\tilde{x} = [x_n^* \dots x_1^*]^T$$

An important property of any Hermitian Toeplitz matrix R is that

$$y = Rx \quad \Rightarrow \quad \tilde{y} = R\tilde{x} \tag{3.5.3}$$

The result (3.5.3) follows from

$$\begin{aligned}\tilde{y}_i = y^*_{n-i+1} &= \sum_{k=1}^{n} R^*_{n-i+1,k} x^*_k \\ &= \sum_{k=1}^{n} \rho^*_{n-i+1-k} x^*_k = \sum_{p=1}^{n} \rho^*_{p-i} x^*_{n-p+1} = \sum_{p=1}^{n} R_{i,p} \tilde{x}_p \\ &= (R\tilde{x})_i\end{aligned}$$

where $R_{i,j}$ denotes the (i,j)th element of the matrix R.

3.5.1 Levinson–Durbin Algorithm

The basic idea of the LDA is to solve (3.4.6) *recursively in* n, starting from the solution for $n = 1$ (which is easily found). By using (3.4.6) and the nested structure of the R matrix, we can write

$$R_{n+2} \begin{bmatrix} 1 \\ \theta_n \\ \hline 0 \end{bmatrix} = \left[\begin{array}{cc|c} & & \rho^*_{n+1} \\ & R_{n+1} & \tilde{r}_n \\ \hline \rho_{n+1} & \tilde{r}^*_n & \rho_0 \end{array}\right] \begin{bmatrix} 1 \\ \theta_n \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \\ \hline \alpha_n \end{bmatrix} \tag{3.5.4}$$

where

$$r_n = [\rho_1 \dots \rho_n]^T \tag{3.5.5}$$

$$\alpha_n = \rho_{n+1} + \tilde{r}^*_n \theta_n \tag{3.5.6}$$

Equation (3.5.4) would be the counterpart of (3.4.6) when n is increased by one, if α_n in (3.5.4) could be nulled. To do so, let

$$k_{n+1} = -\alpha_n / \sigma_n^2 \tag{3.5.7}$$

It follows from (3.5.3) and (3.5.4) that

$$\begin{aligned} R_{n+2} \left\{ \begin{bmatrix} 1 \\ \theta_n \\ 0 \end{bmatrix} + k_{n+1} \begin{bmatrix} 0 \\ \tilde{\theta}_n \\ 1 \end{bmatrix} \right\} &= \begin{bmatrix} \sigma_n^2 \\ 0 \\ \alpha_n \end{bmatrix} + k_{n+1} \begin{bmatrix} \alpha^*_n \\ 0 \\ \sigma_n^2 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_n^2 + k_{n+1}\alpha^*_n \\ 0 \end{bmatrix} \end{aligned} \tag{3.5.8}$$

which has the same structure as

$$R_{n+2}\begin{bmatrix} 1 \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} \sigma_{n+1}^2 \\ 0 \end{bmatrix} \tag{3.5.9}$$

Comparing (3.5.8) with (3.5.9) and making use of the fact that the solution to (3.4.6) is unique for any n, we reach the conclusion that

$$\theta_{n+1} = \begin{bmatrix} \theta_n \\ 0 \end{bmatrix} + k_{n+1}\begin{bmatrix} \tilde{\theta}_n \\ 1 \end{bmatrix} \tag{3.5.10}$$

and

$$\sigma_{n+1}^2 = \sigma_n^2\left(1 - |k_{n+1}|^2\right) \tag{3.5.11}$$

constitute the solution to (3.4.6) for order $(n+1)$.

Equations (3.5.10) and (3.5.11) form the core of the LDA. The initialization of these recursive-in-n equations is straightforward. The following box summarizes the LDA in a form that should be convenient for machine coding. The LDA has many interesting properties and uses, for which we refer to [Söderström and Stoica 1989; Marple 1987; Kay 1988]. The coefficients k_i in the LDA are often called the *reflection coefficients*; $-k_i$ are also called the *partial correlation (PARCOR) coefficients*. The motivation for the name "partial correlation coefficient" is developed in Complement 3.9.1.

The Levinson–Durbin Algorithm

Initialization:

$$\theta_1 = -\rho_1/\rho_0 = k_1 \qquad [1 \text{ flop}]$$

$$\sigma_1^2 = \rho_0 - |\rho_1|^2/\rho_0 \qquad [1 \text{ flop}]$$

For $n = 1, \ldots, n_{\max}$, do:

$$k_{n+1} = -\frac{\rho_{n+1} + \tilde{r}_n^* \theta_n}{\sigma_n^2} \qquad [n+1 \text{ flops}]$$

$$\sigma_{n+1}^2 = \sigma_n^2(1 - |k_{n+1}|^2) \qquad [2 \text{ flops}]$$

$$\theta_{n+1} = \begin{bmatrix} \theta_n \\ 0 \end{bmatrix} + k_{n+1}\begin{bmatrix} \tilde{\theta}_n \\ 1 \end{bmatrix} \qquad [n \text{ flops}]$$

It can be seen from the box that the LDA requires on the order of $2n$ flops to compute $\{\theta_{n+1}, \sigma_{n+1}^2\}$ from $\{\theta_n, \sigma_n^2\}$. Hence, a total of about $n_{\max}^2$ flops is needed to compute all the

solutions to the Yule–Walker system of equations, from $n = 1$ to $n = n_{\max}$. This confirms the claim that the LDA reduces the computational burden associated with a general solver by two orders of magnitude.

3.5.2 Delsarte–Genin Algorithm

In the *real data case* (i.e., whenever $y(t)$ is real valued), the Delsarte–Genin algorithm (DGA), or the *split-Levinson algorithm*, exploits some further structure of the Yule–Walker problem (which is not exploited by the LDA) to decrease even more the number of flops required to solve for $\{\theta_n, \sigma_n^2\}$ [Delsarte and Genin 1986]. In the following, we present a derivation of the DGA that is simpler than the original derivation. As already stated, we assume that the covariance elements $\{\rho_k\}$ in the Yule–Walker equations are real valued.

Let Δ_n be defined by

$$R_{n+1}\Delta_n = \beta_n \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{3.5.12}$$

where the scalar β_n is unspecified for the moment. The matrix R_{n+1} is positive definite, so the $(n+1)$-vector Δ_n is uniquely defined by (3.5.12) (once β_n is specified; as a matter of fact, note that β_n only has a scaling effect on the components of Δ_n). It follows from (3.5.12) and (3.5.3) that Δ_n is a "symmetric vector": It satisfies

$$\Delta_n = \tilde{\Delta}_n \tag{3.5.13}$$

The key idea of the DGA is to introduce such symmetric vectors into the computations involved by the LDA; then only half of the elements of these vectors will need to be computed.

Next, note that, by using the nested structure of R_{n+1} and the defining equation (3.5.12), we can write

$$R_{n+1}\begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix} = \begin{bmatrix} \rho_0 & r_n^T \\ r_n & R_n \end{bmatrix}\begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix} = \begin{bmatrix} \gamma_{n-1} \\ \beta_{n-1} \\ \vdots \\ \beta_{n-1} \end{bmatrix} \tag{3.5.14}$$

where r_n is defined in (3.5.5) and

$$\gamma_{n-1} = r_n^T \Delta_{n-1} \tag{3.5.15}$$

The systems of equations (3.5.12) and (3.5.14) can be combined linearly into a system having the structure of (3.4.6). To do so, let

$$\lambda_n = \beta_n / \beta_{n-1} \tag{3.5.16}$$

Then, from (3.5.12), (3.5.14) and (3.5.16), we get

$$R_{n+1}\left\{\Delta_n - \lambda_n \begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix}\right\} = \begin{bmatrix} \beta_n - \lambda_n \gamma_{n-1} \\ 0 \end{bmatrix} \tag{3.5.17}$$

It will be shown that β_n can always be chosen so as to make the first element of Δ_n equal to 1:

$$(\Delta_n)_1 = 1 \tag{3.5.18}$$

In such a case, (3.5.17) has exactly the same structure as (3.4.6) and, the solutions to these two systems of equations being unique, we are led to the following relations:

$$\begin{bmatrix} 1 \\ \theta_n \end{bmatrix} = \Delta_n - \lambda_n \begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix} \tag{3.5.19}$$

$$\sigma_n^2 = \beta_n - \lambda_n \gamma_{n-1} \tag{3.5.20}$$

Furthermore, $(\Delta_n)_1 = 1$ and Δ_n is a symmetric vector, so we must also have $(\Delta_n)_{n+1} = 1$. This observation, along with (3.5.19) and the fact that k_n is the last element of θ_n (see (3.5.10)), gives the following expression for k_n:

$$k_n = 1 - \lambda_n \tag{3.5.21}$$

The equations (3.5.19)–(3.5.21) express the LDA variables $\{\theta_n, \sigma_n^2, k_n\}$ as functions of $\{\Delta_n\}$ and $\{\beta_n\}$. It remains to derive recursive-in-n formulas for $\{\Delta_n\}$ and $\{\beta_n\}$ and to prove that (3.5.18) really holds. This is done next.

Let $\{\beta_n\}$ be defined recursively by the second-order difference equation

$$\beta_n = 2\beta_{n-1} - \alpha_n \beta_{n-2} \tag{3.5.22}$$

where

$$\alpha_n = (\beta_{n-1} - \gamma_{n-1})/(\beta_{n-2} - \gamma_{n-2}) \tag{3.5.23}$$

The initial values required to start the recursion (3.5.22) are $\beta_0 = \rho_0$ and $\beta_1 = \rho_0 + \rho_1$. With this definition of $\{\beta_n\}$, we claim that the vectors $\{\Delta_n\}$ (as defined in (3.5.12)) satisfy both (3.5.18) and the following second-order recursion:

$$\Delta_n = \begin{bmatrix} \Delta_{n-1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix} - \alpha_n \begin{bmatrix} 0 \\ \Delta_{n-2} \\ 0 \end{bmatrix} \tag{3.5.24}$$

In order to prove the previous claim, we first apply the result (3.5.3) to (3.5.14) to get

$$R_{n+1}\begin{bmatrix}\Delta_{n-1}\\0\end{bmatrix}=\begin{bmatrix}\beta_{n-1}\\\vdots\\\beta_{n-1}\\\gamma_{n-1}\end{bmatrix} \tag{3.5.25}$$

Next, we note that

$$R_{n+1}\begin{bmatrix}0\\\Delta_{n-2}\\0\end{bmatrix}=\begin{bmatrix}\rho_0 & r_{n-1}^T & \rho_n\\ r_{n-1} & R_{n-1} & \tilde{r}_{n-1}\\ \rho_n & \tilde{r}_{n-1}^T & \rho_0\end{bmatrix}\begin{bmatrix}0\\\Delta_{n-2}\\0\end{bmatrix}=\begin{bmatrix}\gamma_{n-2}\\\beta_{n-2}\\\vdots\\\beta_{n-2}\\\gamma_{n-2}\end{bmatrix} \tag{3.5.26}$$

The right-hand sides of equations (3.5.14), (3.5.25), and (3.5.26) can be combined linearly, as described next, to get the right-hand side of (3.5.12):

$$\begin{bmatrix}\gamma_{n-1}\\\beta_{n-1}\\\vdots\\\beta_{n-1}\end{bmatrix}+\begin{bmatrix}\beta_{n-1}\\\vdots\\\beta_{n-1}\\\gamma_{n-1}\end{bmatrix}-\alpha_n\begin{bmatrix}\gamma_{n-2}\\\beta_{n-2}\\\vdots\\\beta_{n-2}\\\gamma_{n-2}\end{bmatrix}=\beta_n\begin{bmatrix}1\\\vdots\\1\end{bmatrix} \tag{3.5.27}$$

The equality in (3.5.27) follows from the defining equations of β_n and α_n. This observation, in conjunction with (3.5.14), (3.5.25) and (3.5.26), gives the system of linear equations

$$R_{n+1}\left\{\begin{bmatrix}\Delta_{n-1}\\0\end{bmatrix}+\begin{bmatrix}0\\\Delta_{n-1}\end{bmatrix}-\alpha_n\begin{bmatrix}0\\\Delta_{n-2}\\0\end{bmatrix}\right\}=\beta_n\begin{bmatrix}1\\\vdots\\1\end{bmatrix} \tag{3.5.28}$$

which has exactly the structure of (3.5.12). Since the solutions to (3.5.12) and (3.5.28) are unique, they must coincide; hence, (3.5.24) follows.

Next, turn to the condition (3.5.18). From (3.5.24), we see that $(\Delta_n)_1 = (\Delta_{n-1})_1$. Hence, in order to prove that (3.5.18) holds, it suffices to show that $\Delta_1 = [1 \quad 1]^T$. The initial values $\beta_0 = \rho_0$ and $\beta_1 = \rho_0 + \rho_1$ (purposely chosen for the sequence $\{\beta_n\}$), when inserted in (3.5.12), give $\Delta_0 = 1$ and $\Delta_1 = [1 \ \ 1]^T$. With this observation, the proof of (3.5.18) and (3.5.24) is finished.

The DGA consists of the equations (3.5.16) and (3.5.19)–(3.5.24). These equations include second-order recursions and appear to be more complicated than the first-order recursive equations of the LDA. In reality, the *symmetry of the* Δ_n *vectors* makes the DGA more efficient computationally than the LDA (as is shown next). The DGA equations are summarized in the next box along with an approximate count of the number of flops required for implementation.

The Delsarte–Genin Algorithm

DGA equations	Operation count: no. of (×)	Operation count: no. of (+)
Initialization:		
$\Delta_0 = 1,\ \beta_0 = \rho_0,\ \gamma_0 = \rho_1$	–	–
$\Delta_1 = [1 \quad 1]^T,\ \beta_1 = \rho_0 + \rho_1,\ \gamma_1 = \rho_1 + \rho_2$	–	2
For $n = 2, \ldots, n_{\max}$, do the following steps:		
(a) $\alpha_n = (\beta_{n-1} - \gamma_{n-1})/(\beta_{n-2} - \gamma_{n-2})$	1	2
$\beta_n = 2\beta_{n-1} - \alpha_n \beta_{n-2}$	2	1
$\Delta_n = \begin{bmatrix} \Delta_{n-1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix} - \alpha_n \begin{bmatrix} 0 \\ \Delta_{n-2} \\ 0 \end{bmatrix}$	$\sim n/2$	$\sim n$
$\gamma_n = r_{n+1}^T \Delta_n = (\rho_1 + \rho_{n+1}) + \Delta_{n,2}(\rho_2 + \rho_n) + \ldots$	$\sim n/2$	$\sim n$
(b) $\lambda_n = \beta_n / \beta_{n-1}$	1	–
$\sigma_n^2 = \beta_n - \lambda_n \gamma_{n-1}$	1	1
$k_n = 1 - \lambda_n$	–	1
(c) $\begin{bmatrix} 1 \\ \theta_n \end{bmatrix} = \Delta_n - \lambda_n \begin{bmatrix} 0 \\ \Delta_{n-1} \end{bmatrix}$	$\sim n/2$	$\sim n$

The DGA can be implemented in two principal modes, to suit the application at hand.

DGA—Mode 1. In most AR modeling exercises, we do not really need all $\{\theta_n\}_{n=1}^{n_{\max}}$. We do, however, need $\{\sigma_1^2, \sigma_2^2, \ldots\}$ for the purpose of order selection (see Appendix C). Let the selected order be denoted by $n_{\max}$. Then the only θ vector to be computed is $\theta_{n_{\max}}$. We might also need to compute the $\{k_n\}$ sequence, because this bears useful information about the stability of the AR model. (See, for example, [Söderström and Stoica 1989; Kay 1988; Therrien 1992].)

In the modeling application we have outlined, we need to iterate only the groups (a) and (b) of equations in the previous DGA summary. The matrix equation (c) is computed only for $n = n_{\max}$. This way of implementing the DGA requires the following number of multiplications and additions:

$$\text{no. of } (\times) \simeq n_{\max}^2/2 \qquad \text{no. of } (+) \simeq n_{\max}^2 \tag{3.5.29}$$

Recall that, for LDA, no. of $(\times)$ = no. of $(+) \simeq n_{\max}^2$. Thus, the DGA is approximately *twice as fast* as the LDA (on computers for which multiplication is much more time consuming than addition). We also remark that, in some parameter-estimation applications, the equations in group (b) of the DGA can also be left out, but doing so will speed up the implementation of the DGA only slightly.

DGA—Mode 2. In other applications, we need all $\{\theta_n\}_{n=1}^{n_{\max}}$. An example of such an application is the Cholesky factorization of the inverse covariance matrix $R_{n_{\max}}^{-1}$. (See, for example, Exercise 3.7 and [SÖDERSTRÖM AND STOICA 1989].) In such a case, we need to iterate all equations in the DGA and so need the following number of arithmetic operations:

$$\text{no. of } (\times) \simeq 0.75n_{\max}^2 \qquad \text{no. of } (+) \simeq 1.5n_{\max}^2 \tag{3.5.30}$$

This is still about 25% faster than the LDA (assuming, once again, that the computation time required for multiplication dominates the time corresponding to an addition).

In closing this section, we note that the computational comparisons between the DGA and the LDA neglected terms on the order $\mathcal{O}(n_{\max})$. This is acceptable if $n_{\max}$ is reasonably large (say, $n_{\max} \geq 10$). If $n_{\max}$ is small, then these comparisons are no longer valid and, in fact, LDA could be more efficient computationally than the DGA in such a case. In such low-dimensional applications, the LDA is therefore to be preferred to the DGA. Also recall that the LDA is the algorithm to use with complex-valued data; the DGA does not appear to have a computationally efficient extension for complex-valued data.

3.6 MA SIGNALS

According to the definition in (3.2.8), an *MA signal* is obtained by filtering white noise with an *all-zero filter*. This all-zero structure makes it impossible to use an MA equation to model a spectrum with sharp peaks unless the MA order is chosen "sufficiently large." This is to be contrasted with the ability of the AR (or "all-pole") equation to model narrowband spectra by using fairly low model orders (per the discussion in the previous sections). The MA model provides a good approximation for those spectra characterized by broad peaks and sharp nulls. Such spectra are encountered less frequently in applications than are narrowband spectra, so there is a somewhat limited engineering interest in using the MA signal model for spectral estimation. Another reason for this limited interest is that the MA parameter-estimation problem is basically a nonlinear one and is significantly more difficult to solve than the AR parameter-estimation problem. In any case, the types of difficulties we must face in MA and ARMA estimation problems are quite similar; hence, we almost always prefer to use the more general ARMA model in lieu of the MA one. For these reasons, our discussion of MA spectral estimation will be brief.

One method for estimating an MA spectrum consists of two steps: (i) Estimate the MA parameters $\{b_k\}_{k=1}^m$ and σ^2; and (ii) insert the estimated parameters from the first step in the MA PSD formula (see (3.2.2)). The result is

$$\hat{\phi}(\omega) = \hat{\sigma}^2 |\hat{B}(\omega)|^2 \tag{3.6.1}$$

The difficulty with this approach lies in step (i), which is a nonlinear estimation problem. Approximate linear solutions to this problem do, however, exist. One of these approximate procedures, perhaps the method most used for MA parameter estimation, is based on a two-stage least-squares methodology [DURBIN 1959]. It is called *Durbin's method*; it will be described in Section 3.7 in the more general context of ARMA parameter estimation.

Another method to estimate an MA spectrum is based on the reparameterization of the PSD in terms of the covariance sequence. We see from (3.2.8) that, for an MA of order m,

$$r(k) = 0 \qquad \text{for } |k| > m \tag{3.6.2}$$

This simple observation turns the definition of the PSD as a function of $\{r(k)\}$ into a finite-dimensional spectral model:

$$\phi(\omega) = \sum_{k=-m}^{m} r(k) e^{-i\omega k} \tag{3.6.3}$$

Hence, a simple estimator of MA PSD is obtained by inserting estimates of $\{r(k)\}_{k=0}^{m}$ in (3.6.3). If the standard sample covariances $\{\hat{r}(k)\}$ are used to estimate $\{r(k)\}$, then we obtain

$$\hat{\phi}(\omega) = \sum_{k=-m}^{m} \hat{r}(k) e^{-i\omega k} \tag{3.6.4}$$

This spectral estimate is of the form of the Blackman–Tukey estimator (2.5.1). More precisely, (3.6.4) coincides with a Blackman–Tukey estimator using a rectangular window of length $2m + 1$. This is not unexpected. If we impose the zero-bias restriction on the nonparametric approach to spectral estimation (to make the comparison with the parametric approach fair), then the Blackman–Tukey estimator with a rectangular window of length $2m + 1$ implicitly assumes that the covariance lags outside the window interval are equal to zero. This is precisely the assumption behind the MA signal model; see (3.6.2). Alternatively, if we make use of the assumption (3.6.2) in a Blackman–Tukey estimator, then we definitely end up with (3.6.4), as, in such a case, this is the spectral estimator in the Blackman–Tukey class with zero bias and "minimum" variance.

The analogy between the Blackman–Tukey and MA spectrum estimation methods makes it simpler to understand a problem associated with the MA spectral estimator (3.6.4). The (implicit) use of a rectangular window in (3.6.4) means that the spectral estimate so obtained is not necessarily positive at all frequencies (see (2.5.5) and the discussion following that equation). Indeed, it is often noted in applications that (3.6.4) produces PSD estimates that are negative at some frequencies. In order to cure this deficiency of (3.6.4), we may use another lag window in lieu of the rectangular one—one guaranteed to be positive semidefinite. This way of correcting $\hat{\phi}(\omega)$ in (3.6.4) is, of course, reminiscent of the Blackman–Tukey approach. It should be noted, however, that $\hat{\phi}(\omega)$, when thus corrected, is no longer an unbiased estimator of the PSD of an MA(m) signal. (See, for example, [Moses and Beex 1986] for details on this aspect.)

3.7 ARMA SIGNALS

Spectra with both sharp peaks and deep nulls cannot be modeled by either AR or MA equations of reasonably small orders. There are, of course, other instances of rational spectra that cannot

be described exactly as AR or MA spectra. It is in these cases that the more general *ARMA model*, also called the *pole–zero model*, is valuable. However, the great initial promise of ARMA spectral estimation diminishes to some extent because there is yet no well-established algorithm, from both theoretical and practical standpoints, for ARMA parameter estimation. The "theoretically optimal ARMA estimators" are based on iterative procedures whose global convergence is not guaranteed. The "practical ARMA estimators," on the other hand, are computationally simple and often quite reliable, but their statistical accuracy is in some cases poor. In the following, we describe two ARMA spectral estimation algorithms that have been used in applications with a reasonable degree of success. See also [Byrnes, Georgiou, and Lindquist 2000; Byrnes, Georgiou, and Lindquist 2001] for some recent results on ARMA parameter estimation.

3.7.1 Modified Yule–Walker Method

The modified Yule–Walker method is a two-stage procedure for estimating the ARMA spectral density. In the first stage, we estimate the AR coefficients by using equation (3.3.4). In the second stage, we use the AR coefficient and ACS estimates in equation (3.2.1) to estimate the γ_k coefficients. We describe these two stages in this section.

Writing equation (3.3.4) for $k = m+1, m+2, \ldots, m+M$ in a matrix form gives

$$\begin{bmatrix} r(m) & r(m-1) & \ldots & r(m-n+1) \\ r(m+1) & r(m) & & r(m-n+2) \\ \vdots & & \ddots & \vdots \\ r(m+M-1) & \ldots & \ldots & r(m-n+M) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = - \begin{bmatrix} r(m+1) \\ r(m+2) \\ \vdots \\ r(m+M) \end{bmatrix} \tag{3.7.1}$$

If we set $M = n$ in (3.7.1), we obtain a system of n equations in n unknowns. This constitutes a generalization of the Yule–Walker system of equations that holds in the AR case. Hence, these equations are said to form the *modified Yule–Walker* (MYW) system of equations [Gersh 1970; Kinkel, Perl, Scharf, and Stubberud 1979; Beex and Scharf 1981; Cadzow 1982]. Replacing the theoretical covariances $\{r(k)\}$ by their sample estimates $\{\hat{r}(k)\}$ in these equations leads to

$$\boxed{\begin{bmatrix} \hat{r}(m) & \ldots & \hat{r}(m-n+1) \\ \vdots & & \vdots \\ \hat{r}(m+n-1) & \ldots & \hat{r}(m) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_n \end{bmatrix} = - \begin{bmatrix} \hat{r}(m+1) \\ \vdots \\ \hat{r}(m+n) \end{bmatrix}} \tag{3.7.2}$$

This linear system can be solved for $\hat{a}_1, \ldots, \hat{a}_n$, which are called the *modified Yule–Walker estimates* of $a_1, \ldots, a_n$. The square matrix in (3.7.2) can be shown to be nonsingular under mild conditions. There exist fast algorithms of the Levinson type for solving *non-Hermitian* Toeplitz systems of equations of the form of (3.7.2); they require about twice the computation of the LDA algorithm. See [Marple 1987; Kay 1988; Söderström and Stoica 1989].

The MYW AR estimate has reasonable accuracy if the zeroes of $B(z)$ in the ARMA model are well inside the unit circle. However, (3.7.2) could give very inaccurate estimates in those

cases where the poles and zeroes of the ARMA model description are closely spaced together at positions near the unit circle. Such ARMA models, with nearly coinciding poles and zeroes of modulus close to one, correspond to narrowband signals. The covariance sequence of narrowband signals decays very slowly. Indeed, as we know, the more concentrated a signal is in frequency, usually the more expanded it is in time, and vice versa. This means that there is "information" in the higher lag covariances of the signal that can be exploited to improve the accuracy of the AR coefficient estimates. We can exploit the additional information by choosing $M > n$ in equation (3.7.1) and solving the overdetermined system of equations so obtained. If we replace the true covariances in (3.7.1) with $M > n$ by finite-sample estimates, there will in general be no exact solution. A natural idea to overcome this problem is to solve the resultant equation

$$\hat{R}\hat{a} \simeq -\hat{r} \tag{3.7.3}$$

in a least-squares (LS) or total-least-squares (TLS) sense (see Appendix A). Here, $\hat{R}$ and $\hat{r}$ represent the ACS matrix and vector in (3.7.1) with sample ACS estimates replacing the true ACS there. For instance, the (weighted) least-squares solution to (3.7.3) is, mathematically, given by[1]

$$\hat{a} = -(\hat{R}^* W \hat{R})^{-1}(\hat{R}^* W \hat{r}) \tag{3.7.4}$$

where W is an $M \times M$ positive definite weighting matrix. The AR estimate derived from (3.7.3) with $M > n$ is called the *overdetermined modified YW estimate* [Beex and Scharf 1981; Cadzow 1982].

Some notes on the choice between (3.7.2) and (3.7.3), and on the selection of M, are in order.

- Choosing $M > n$ does not always improve the accuracy of the previous AR coefficient estimates. In fact, if the poles and zeroes are not close to the unit circle, choosing $M > n$ can make the accuracy *worse*. When the ACS decays slowly to zero, however, choosing $M > n$ generally improves the accuracy of $\hat{a}$ [Cadzow 1982; Stoica, Friedlander, and Söderström 1987b]. A qualitative explanation for this phenomenon can be seen by thinking of a finite-sample ACS estimate as being the sum of its "signal" component $r(k)$ and a "noise" component due to finite-sample estimation: $\hat{r}(k) = r(k) + n(k)$. If the ACS decays slowly to zero, the signal component is "large" compared to the noise component, even for relatively large values of k, and including $\hat{r}(k)$ in the estimation of $\hat{a}$ improves accuracy. If the noise component of $\hat{r}(k)$ dominates, including $\hat{r}(k)$ in the estimation of $\hat{a}$ could decrease the accuracy of $\hat{a}$.
- The *statistical and numerical accuracies* of the solution $\{\hat{a}_i\}$ to (3.7.3) are quite interrelated. In more exact but still loose terms, it can be shown that the statistical accuracy of $\{\hat{a}_i\}$ is poor (good) if the condition number of the matrix $\hat{R}$ in (3.7.3) is large (small). (See [Stoica, Friedlander, and Söderström 1987b; Söderström and Stoica 1989] and also Appendix A.) This observation suggests that M should be selected so as to make the matrix in (3.7.3) reasonably well conditioned. In order to make a connection between this rule of

[1] From a numerical viewpoint, equation (3.7.4) is not a particularly good way to solve (3.7.3). A more numerically sound approach is to use the QR decomposition; see Section A.8.2 for details.

thumb for selecting M and the previous explanation for the poor accuracy of (3.7.2) in the case of narrowband signals, note that, for slowly decaying covariance sequences, the columns of the matrix in (3.7.2) are nearly linearly dependent. Hence, the condition number of the covariance matrix could be quite high in such a case, and we might need to increase M in order to lower the condition number to a reasonable value.

- The weighting matrix W in (3.7.4) can also be chosen to improve the accuracy of the AR coefficient estimates. A simple first choice is $W = I$, resulting in the regular (unweighted) least squares estimate. Some accuracy improvement can be obtained by choosing W to be diagonal with decreasing positive diagonal elements (to reflect the decreased confidence in higher ACS lag estimates). In addition, optimal weighting matrices have been derived (see [Stoica, Friedlander, and Söderström 1987a]); the optimal weight minimizes the covariance of $\hat{a}$ (for large N) over all choices of W. Unfortunately, the optimal weight depends on the (unknown) ARMA parameters. Thus, to use optimally weighted methods, a two-step "bootstrap" approach is used, in which a fixed W is first chosen and initial parameter estimates are obtained; these initial estimates are used to form an optimal W, and a second estimation gives the "optimal accuracy" AR coefficients. As a general rule, the performance gain from using optimal weighting is relatively small compared to the computational overhead required to compute the optimal weighting matrix. Most of the accuracy improvement can be realized by choosing $M > n$ and $W = I$ for many problems. We refer the reader to [Stoica, Friedlander, and Söderström 1987a; Cadzow 1982] for a discussion on the effect of W on the accuracy of $\hat{a}$ and on optimal weighting matrices.

Once the AR estimates are obtained, we turn to the problem of estimating the MA part of the ARMA spectrum. Let

$$\gamma_k = E\left\{[B(z)e(t)][B(z)e(t-k)]^*\right\} \tag{3.7.5}$$

denote the covariances of the MA part. Since the PSD of this part of the ARMA signal model is given by (see (3.6.1) and (3.6.3))

$$\sigma^2|B(\omega)|^2 = \sum_{k=-m}^{m} \gamma_k e^{-i\omega k} \tag{3.7.6}$$

it suffices to estimate $\{\gamma_k\}$ in order to characterize the spectrum of the MA part. From (3.2.7) and (3.7.5), we obtain

$$\begin{aligned}
\gamma_k &= E\left\{[A(z)y(t)][A(z)y(t-k)]^*\right\} \\
&= \sum_{j=0}^{n}\sum_{p=0}^{n} a_j a_p^* E\left\{y(t-j)y^*(t-k-p)\right\} \\
&= \sum_{j=0}^{n}\sum_{p=0}^{n} a_j a_p^* r(k+p-j) \qquad (a_0 \triangleq 1)
\end{aligned} \tag{3.7.7}$$

for $k = 0, \ldots, m$. Inserting the previously calculated estimates of $\{a_k\}$ and $\{r_k\}$ in (3.7.7) leads to the following estimator of $\{\gamma_k\}$:

$$\hat{\gamma}_k = \begin{cases} \sum_{j=0}^{n} \sum_{p=0}^{n} \hat{a}_j \hat{a}_p^* \hat{r}(k+p-j), & k = 0, \ldots, m \ (\hat{a}_0 \triangleq 1) \\ \hat{\gamma}_{-k}^*, & k = -1, \ldots, -m \end{cases} \tag{3.7.8}$$

Finally, the ARMA spectrum is estimated as follows:

$$\hat{\phi}(\omega) = \frac{\sum_{k=-m}^{m} \hat{\gamma}_k e^{-i\omega k}}{|\hat{A}(\omega)|^2} \tag{3.7.9}$$

The *MA estimate* used by the ARMA spectral estimator in (3.7.9) is of the type (3.6.4) encountered in the MA context. Hence, the criticism of (3.6.4) in the previous section is still valid. In particular, the numerator in (3.7.9) is not guaranteed to be positive for all ω values, so this approach could lead to negative ARMA spectral estimates. See, for example, [Kinkel, Perl, Scharf, and Stubberud 1979; Moses and Beex 1986].

Since (3.7.9) relies on the modified YW method of AR parameter estimation, we call (3.7.9) the *modified YW ARMA spectral estimator*. Refined versions of this ARMA spectral estimator, which improve the estimation accuracy if N is sufficiently large, were proposed in [Stoica and Nehorai 1986; Stoica, Friedlander, and Söderström 1987a; Moses, Šimonytė, Stoica, and Söderström 1994]. A related ARMA spectral estimation method is outlined in Exercise 3.14.

3.7.2 Two-Stage Least-Squares Method

If the noise sequence $\{e(t)\}$ were known, then the problem of estimating the parameters in the ARMA model (3.2.7) would have been a simple *input–output system parameter estimation* problem, which could be solved by several methods, the simplest of which is the *least-squares (LS) method*. In the LS method, we express equation (3.2.7) as

$$y(t) + \varphi^T(t)\theta = e(t) \tag{3.7.10}$$

where

$$\varphi^T(t) = [y(t-1), \ldots, y(t-n) | -e(t-1), \ldots, -e(t-m)]$$
$$\theta = [a_1, \ldots, a_n | b_1, \ldots, b_m]^T$$

Writing (3.7.10) in matrix form for $t = L+1, \ldots, N$ (for some $L > \max(m, n)$) gives

$$z + Z\theta = e \tag{3.7.11}$$

where

$$Z = \left[\begin{array}{ccc|ccc} y(L) & \dots & y(L-n+1) & -e(L) & \dots & -e(L-m+1) \\ y(L+1) & \dots & y(L-n+2) & -e(L+1) & \dots & -e(L-m+2) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(N-1) & \dots & y(N-n) & -e(N-1) & \dots & -e(N-m) \end{array}\right] \tag{3.7.12}$$

$$z = [y(L+1), y(L+2), \dots, y(N)]^T \tag{3.7.13}$$

$$e = [e(L+1), e(L+2), \dots, e(N)]^T \tag{3.7.14}$$

Assume we know Z; then we could solve for θ in (3.7.11) by minimizing $\|e\|^2$. This leads to a least-squares estimate similar to the AR LS estimate introduced in Section 3.4.2 (see also Result R32 in Appendix A):

$$\hat{\theta} = -(Z^*Z)^{-1}(Z^*z) \tag{3.7.15}$$

Of course, the $\{e(t)\}$ in Z are not known. However, they can be estimated as described next.

Since the ARMA model (3.2.7) is of *minimum phase*, by assumption, it can alternatively be written as the infinite-order AR equation

$$(1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \dots)y(t) = e(t) \tag{3.7.16}$$

where the coefficients $\{\alpha_k\}$ of $1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots \triangleq A(z)/B(z)$ converge to zero as k increases. An idea to estimate $\{e(t)\}$ is to first estimate the AR parameters $\{\alpha_k\}$ in (3.7.16) and next obtain $\{e(t)\}$ by filtering $\{y(t)\}$ as in (3.7.16). Of course, we cannot estimate an infinite number of (independent) parameters from a finite number of samples. In practice, the AR equation must be approximated by one of, say, order K. The parameters in the *truncated AR model* of $y(t)$ can be estimated by using either the YW or the LS procedure in Section 3.4.

This discussion leads to the two-stage LS algorithm summarized in the accompanying box. The two-stage LS parameter estimator is also discussed, for example, in [Mayne and Firoozan 1982; Söderström and Stoica 1989]. The spectral estimate is guaranteed to be positive for all frequencies, by construction. Owing to the practical requirement to truncate the AR model (3.7.16), the two-stage LS estimate is biased. The bias can be made small by choosing K sufficiently large; however, K should not be too large with respect to N, or the accuracy of $\hat{\theta}$ in Step 2 will decrease. The difficult case for this method is apparently that of ARMA signals with *zeroes close to the unit circle*. In such a case, it might be necessary to select a very large value of K in order to keep the approximation (bias) errors in Step 1 at a reasonable level. The computational burden of Step 1 could then become prohibitively large. It should be noted, however, that the case of ARMA signals with zeroes near the unit circle is a difficult one for all known ARMA estimation methods [Kay 1988; Marple 1987; Söderström and Stoica 1989].

The Two-Stage Least-Squares ARMA Method

Step 1. Estimate the parameters $\{\alpha_k\}$ in an AR(K) model of $y(t)$ by the YW or covariance LS method. Let $\{\hat{\alpha}_k\}_{k=1}^{K}$ denote the estimated parameters.

Obtain an estimate of the noise sequence $\{e(t)\}$ by

$$\hat{e}(t) = y(t) + \sum_{k=1}^{K} \hat{\alpha}_k y(t-k) \tag{3.7.17}$$

for $t = K+1, \ldots, N$.

Step 2. Replace the $e(t)$ in (3.7.12) by the $\hat{e}(t)$ computed in Step 1. Obtain $\hat{\theta}$ from (3.7.15) with $L = K + m$. Estimate

$$\hat{\sigma}^2 = \frac{1}{N-L}\tilde{e}^*\tilde{e} \tag{3.7.18}$$

where $\tilde{e} = Z\hat{\theta} + z$ is the LS error from (3.7.11).

Insert $\{\hat{\theta}, \hat{\sigma}^2\}$ into the PSD expression (3.2.2) to estimate the ARMA spectrum.

Finally, we remark that the two-stage LS algorithm may be modified to estimate the parameters in MA models, by simply skipping over the estimation of AR parameters in Step 2. This approach was suggested for the first time in [Durbin 1959] and is often called *Durbin's method*.

3.8 MULTIVARIATE ARMA SIGNALS

The multivariate analog of the ARMA signal in equation (3.2.7) is

$$A(z)y(t) = B(z)e(t) \tag{3.8.1}$$

where $y(t)$ and $e(t)$ are $ny \times 1$ vectors, and $A(z)$ and $B(z)$ are $ny \times ny$ matrix polynomials in the unit delay operator. The task of estimating the matrix coefficients—$\{A_i, B_j\}$—of the AR and MA polynomials in (3.8.1) is much more complicated than in the scalar case, for at least one reason: The representation of $y(t)$ in (3.8.1), with all elements in $\{A_i, B_j\}$ assumed to be unknown, could well be *nonunique*, even when the orders of $A(z)$ and $B(z)$ have been chosen correctly. More precisely, assume that we are given the spectral density matrix of an ARMA signal $y(t)$ along with the (minimal) orders of the AR and MA polynomials in its ARMA equation. If all elements of $\{A_i, B_j\}$ are considered to be unknown, then, *unlike in the scalar case*, the previous information could be insufficient for determining the matrix coefficients $\{A_i, B_j\}$ uniquely.

(See, for example, [HANNAN AND DEISTLER 1988] and also Exercise 3.16.) The lack of uniqueness of the representation could lead to a *numerically ill-conditioned parameter-estimation* problem. For instance, this would be the case with the multivariate analog of the *modified Yule–Walker* method discussed in Section 3.7.1.

Apparently, the only possible cure for the aforementioned problem consists of using a *canonical parameterization* for the AR and MA coefficients. Basically, this amounts to setting some of the elements of $\{A_i, B_j\}$ to known values, such as 0 or 1, thereby reducing the number of unknowns. The problem, however, is that, to know which elements should be set to 0 or 1 in a specific case, we need to know *ny indices* (called "structure indices"), which are usually difficult to obtain in practice [KAILATH 1980; HANNAN AND DEISTLER 1988]. The difficulty in obtaining those indices has hampered the use of canonical parameterizations in applications. For this reason, we do not go into any of the details of the canonical forms for ARMA signals. The nonuniqueness of the fully parameterized ARMA equation will, however, receive further attention in Section 3.8.2.

Concerning the other approach to ARMA parameter estimation discussed in Section 3.7.2, namely *the two-stage least-squares method*, it is worth noting that it *can be extended to the multivariate case in a straightforward manner*. In particular, there is *no* need for using a canonical parameterization in either step of the extended method. (See, for example, [SÖDERSTRÖM AND STOICA 1989].) Working the details of the extension is left as an interesting exercise for the reader. We stress that the two-stage LS approach is perhaps the only real competitor to the subspace ARMA parameter-estimation method described in the next subsections.

3.8.1 ARMA State–Space Equations

The difference-equation representation in (3.8.1) can be transformed into the following *state–space representation*, and vice versa (see, for example, [AOKI 1987; KAILATH 1980]):

$$\boxed{\begin{aligned} x(t+1) &= Ax(t) + Be(t) \qquad (n \times 1) \\ y(t) &= Cx(t) + e(t) \qquad (ny \times 1) \end{aligned}} \tag{3.8.2}$$

Thereafter, $x(t)$ is the state vector of dimension n; A, B, and C are matrices of appropriate dimensions (with A having all eigenvalues inside the unit circle); and $e(t)$ is white noise with zero mean and with covariance matrix denoted by Q. We thus have

$$E\{e(t)\} = 0 \tag{3.8.3}$$

$$E\left\{e(t)e^*(s)\right\} = Q\delta_{t,s} \tag{3.8.4}$$

where Q is positive definite by assumption.

The *transfer filter* corresponding to (3.8.2), also called *the ARMA shaping filter*, is readily seen to be

$$H(z) = z^{-1}C(I - Az^{-1})^{-1}B + I \tag{3.8.5}$$

By paralleling the calculation leading to (1.4.9), it is then possible to show that the *ARMA power spectral density (PSD) matrix* is given by

$$\boxed{\phi(\omega) = H(\omega)QH^*(\omega)} \tag{3.8.6}$$

The derivation of (3.8.6) is left as an exercise for the reader.

In the next subsections, we will introduce a methodology for estimating the matrices A, B, C, and Q of the state-space equation (3.8.2) and, hence, the ARMA power spectral density (via (3.8.5) and (3.8.6)). In this subsection, we derive a number of results that lay the groundwork for the discussion in the next subsections.

Let

$$R_k = E\left\{y(t)y^*(t-k)\right\} \tag{3.8.7}$$

$$P = E\left\{x(t)x^*(t)\right\} \tag{3.8.8}$$

Observe that, for $k \geq 1$,

$$\begin{aligned} R_k &= E\left\{[Cx(t+k) + e(t+k)][x^*(t)C^* + e^*(t)]\right\} \\ &= CE\left\{x(t+k)x^*(t)\right\}C^* + CE\left\{x(t+k)e^*(t)\right\} \end{aligned} \tag{3.8.9}$$

From equation (3.8.2), we obtain (by induction)

$$x(t+k) = A^k x(t) + \sum_{\ell=0}^{k-1} A^{k-\ell-1}\, Be(t+\ell) \tag{3.8.10}$$

which implies that

$$E\left\{x(t+k)x^*(t)\right\} = A^k P \tag{3.8.11}$$

and

$$E\left\{x(t+k)e^*(t)\right\} = A^{k-1}BQ \tag{3.8.12}$$

Inserting (3.8.11) and (3.8.12) into (3.8.9) yields

$$\boxed{R_k = CA^{k-1}D \qquad (\text{for } k \geq 1)} \tag{3.8.13}$$

where

$$\boxed{D = APC^* + BQ} \tag{3.8.14}$$

From the first equation in (3.8.2), we also readily obtain

$$P = APA^* + BQB^* \tag{3.8.15}$$

and, from the second equation,

$$R_0 = CPC^* + Q \tag{3.8.16}$$

It follows from (3.8.14) and (3.8.16) that

$$\boxed{B = (D - APC^*)Q^{-1}} \tag{3.8.17}$$

and, respectively,

$$\boxed{Q = R_0 - CPC^*} \tag{3.8.18}$$

Finally, inserting (3.8.17) and (3.8.18) into (3.8.15) gives the following *Riccati equation* for P:

$$\boxed{P = APA^* + (D - APC^*)(R_0 - CPC^*)^{-1}(D - APC^*)^*} \tag{3.8.19}$$

The results lead to a number of interesting observations.

The (Non)Uniqueness Issue. It is well known that a linear nonsingular transformation of the state vector in (3.8.2) leaves the transfer-function matrix associated with (3.8.2) unchanged. To be more precise, let the new state vector be given by

$$\tilde{x}(t) = Tx(t), \qquad (|T| \neq 0) \tag{3.8.20}$$

It can be verified that the state–space equations in $\tilde{x}(t)$, corresponding to (3.8.2), are

$$\begin{aligned} \tilde{x}(t+1) &= \tilde{A}\tilde{x}(t) + \tilde{B}e(t) \\ y(t) &= \tilde{C}\tilde{x}(t) + e(t) \end{aligned} \tag{3.8.21}$$

where

$$\tilde{A} = TAT^{-1}; \quad \tilde{B} = TB; \quad \tilde{C} = CT^{-1} \tag{3.8.22}$$

As $\{y(t)\}$ and $\{e(t)\}$ in (3.8.21) are the same as in (3.8.2), the transfer function $H(z)$ from $e(t)$ to $y(t)$ must be the same for both (3.8.2) and (3.8.21). (Verifying this by direct calculation is left to the reader.) The consequence is that there exists an *infinite number* of triples (A, B, C) (with *all* matrix elements assumed unknown) that lead to the same ARMA transfer function and, hence, the same ARMA covariance sequence and PSD matrix. For the transfer-function matrix, the nonuniqueness induced by the *similarity transformation* (3.8.22) is the only type possible (as we know from the deterministic system theory, for example, [Kailath 1980]). For the covariance sequence and the PSD, however, other types of nonuniqueness are also possible. See, for example, [Faurre 1976] and [Söderström and Stoica 1989, Problem 6.3].

Most ARMA estimation methods require the use of a uniquely parameterized representation. The previous discussion has clearly shown that letting all elements of A, B, C, and Q be unknown does not lead to such a unique representation. The latter representation is obtained only if a

canonical form is used. As already explained, the ARMA parameter estimation methods relying on canonical parameterizations are impractical. The subspace-based estimation approach discussed in the next subsection circumvents the canonical-parameterization requirement in an interesting way: The nonuniqueness of the ARMA representation with A, B, C, and Q fully parameterized is reduced to the nonuniqueness of a certain decomposition of covariance matrices; then by choosing a specific decomposition, a triplet (A, B, C) is isolated and is determined in a numerically well-posed manner.

The Minimality Issue. Let, for some integer-valued m,

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{m-1} \end{bmatrix} \tag{3.8.23}$$

and

$$\mathcal{C}^* = [D \; AD \; \cdots \; A^{m-1}D] \tag{3.8.24}$$

The similarity between the above matrices and the *observability and controllability matrices*, respectively, from the theory of deterministic state–space equations is evident. In fact, it follows from the aforementioned theory and from (3.8.13) that the triplet (A, D, C) is a *minimal representation* (i.e., one with the minimum possible dimension n) *of the covariance sequence* $\{R_k\}$ if and only if. (See, for example, [Kailath 1980; Hannan and Deistler 1988].)

$$\boxed{\operatorname{rank}(\mathcal{O}) = \operatorname{rank}(\mathcal{C}) = n \qquad (\text{for } m \geq n)} \tag{3.8.25}$$

As was shown previously, the other matrices P, Q, and B of the state–space equation (3.8.2) can be obtained from A, C, and D (see equations (3.8.19), (3.8.18), and (3.8.17), respectively). It follows that the state–space equation (3.8.2) is a minimal representation of the ARMA covariance sequence $\{R_k\}$ if and only if the condition (3.8.25) is satisfied. In what follows, we assume that the "minimality condition" (3.8.25) holds true.

3.8.2 Subspace Parameter Estimation—Theoretical Aspects

We begin by showing how A, C, and D can be obtained from a sequence of theoretical ARMA covariances. Let

$$R = \begin{bmatrix} R_1 & R_2 & \cdots & R_m \\ R_2 & R_3 & \cdots & R_{m+1} \\ \vdots & \vdots & & \vdots \\ R_m & R_{m+1} & \cdots & R_{2m-1} \end{bmatrix}$$

$$= E\left\{ \begin{bmatrix} y(t) \\ \vdots \\ y(t+m-1) \end{bmatrix} [y^*(t-1) \cdots y^*(t-m)] \right\} \tag{3.8.26}$$

denote the *block-Hankel matrix* of covariances. (The name given to (3.8.26) is due to its special structure: the submatrices on its block antidiagonals are identical. Such a matrix is a block extension to the standard Hankel matrix; see Definition D14 in Appendix A.) According to (3.8.13), we can factor R as follows:

$$R = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{m-1} \end{bmatrix} [D \; AD \; \cdots A^{m-1} \; D] = \mathcal{O}\mathcal{C}^* \tag{3.8.27}$$

It follows from (3.8.25) and (3.8.27) (see Result R4 in Appendix A) that

$$\text{rank}(R) = n \qquad (\text{for } m \geq n) \tag{3.8.28}$$

Hence, n could, in principle, be obtained as the rank of R. To determine A, C, and D, let us consider the singular value decomposition (SVD) of R (see Appendix A);

$$R = U \Sigma V^* \tag{3.8.29}$$

where Σ is a nonsingular $n \times n$ diagonal matrix, and

$$U^*U = V^*V = I \qquad (n \times n)$$

By comparing (3.8.27) and (3.8.29), we obtain

$$\mathcal{O} = U \Sigma^{1/2} T \quad \text{for some nonsingular transformation matrix } T \tag{3.8.30}$$

because the columns of both $\mathcal{O}$ and $U\Sigma^{1/2}$ are bases of the range space of R. Henceforth, $\Sigma^{1/2}$ denotes a square root of Σ (that is, $\Sigma^{1/2}\Sigma^{1/2} = \Sigma$). By inserting (3.8.30) in the equation $\mathcal{O}\mathcal{C}^* = U\Sigma V^*$, we also obtain

$$\mathcal{C} = V \Sigma^{1/2} (T^{-1})^* \tag{3.8.31}$$

Next, observe that

$$\mathcal{O}T^{-1} = \begin{bmatrix} (CT^{-1}) \\ (CT^{-1})(TAT^{-1}) \\ \vdots \\ (CT^{-1})(TAT^{-1})^{m-1} \end{bmatrix} \tag{3.8.32}$$

and

$$T\mathcal{C}^* = [(TD) \cdots (TAT^{-1})^{m-1}(TD)] \tag{3.8.33}$$

This implies that, *by identifying $\mathcal{O}$ and $\mathcal{C}$ with the matrices made from all possible bases of the range spaces of R and R^*, respectively, we obtain the set of similarity-equivalent triples* (A, D, C).

Hence, picking up a certain basis yields a specific triple (A, D, C) in the aforementioned set. *This is how the subspace approach to ARMA state–space parameter estimation circumvents the nonuniqueness problem associated with a fully parameterized model.*

In view of the previous discussion, we can, for instance, set $T = I$ in (3.8.30) and (3.8.31) and obtain C as the first ny rows of $U\Sigma^{1/2}$ and D as the first ny columns of $\Sigma^{1/2}V^*$. Then, A may be obtained as the solution to the linear system of equations

$$(\bar{U}\Sigma^{1/2})A = \underline{U}\Sigma^{1/2} \tag{3.8.34}$$

where $\bar{U}$ and $\underline{U}$ are the matrices made from the first and, respectively, the last $(m-1)$ block rows of U. Once A, C, and D have been found, P is obtained by solving the Riccati equation (3.8.19), and then Q and B are derived from (3.8.18) and (3.8.17). Algorithms for solving the Riccati equation are presented, for instance, in [VAN OVERSCHEE AND DE MOOR 1996] and the references therein.

A modification of the preceding procedure that does not change the solution obtained in the theoretical case, but appears to have *beneficial effects on the parameter estimates obtained from finite samples*, is as follows: Let us denote the two vectors appearing in (3.8.26) by the symbols

$$f(t) = [y^T(t) \cdots y^T(t+m-1)]^T \tag{3.8.35}$$

and

$$p(t) = [y^T(t-1) \cdots y^T(t-m)]^T \tag{3.8.36}$$

Let

$$R_{fp} = E\left\{f(t)p^*(t)\right\} \tag{3.8.37}$$

and let R_{ff} and R_{pp} be similarly defined. Redefine the matrix in (3.8.26) as

$$R = R_{ff}^{-1/2} R_{fp} R_{pp}^{-1/2} \tag{3.8.38}$$

where $R_{ff}^{-1/2}$ and $R_{pp}^{-1/2}$ are the Hermitian square roots of R_{ff}^{-1} and R_{pp}^{-1}. (See Definition D12 in Appendix A.) A heuristic explanation of why the previous modification should lead to better parameter estimates in finite samples is as follows: The matrix R in (3.8.26) is equal to R_{fp}, whereas the R in (3.8.38) can be written as $R_{\tilde{f}\tilde{p}}$, where both $\tilde{f}(t) = R_{ff}^{-1/2} f(t)$ and $\tilde{p}(t) = R_{pp}^{-1/2} p(t)$ have unity covariance matrices. Owing to the latter property, the cross-covariance matrix $R_{\tilde{f}\tilde{p}}$ and its singular elements are usually estimated more accurately from finite samples than are R_{fp} and its singular elements. This fact should eventually lead to better parameter estimates.

By making use of the factorization (3.8.27) of R_{fp} along with the formula (3.8.38) for the matrix R, we can write

$$R = R_{ff}^{-1/2} R_{fp} R_{pp}^{-1/2} = R_{ff}^{-1/2} \mathcal{O}\mathcal{C}^* R_{pp}^{-1/2} = U\Sigma V^* \tag{3.8.39}$$

where $U\Sigma V^*$ is now the SVD of R in (3.8.38). Identifying $R_{ff}^{-1/2}\mathcal{O}$ with $U\Sigma^{1/2}$ and $R_{pp}^{-1/2}\mathcal{C}$ with $V\Sigma^{1/2}$, we obtain

$$\mathcal{O} = R_{ff}^{1/2} U \Sigma^{1/2} \tag{3.8.40}$$

$$\mathcal{C} = R_{pp}^{1/2} V \Sigma^{1/2} \tag{3.8.41}$$

The matrices A, C, and D can be determined from these equations as previously described. Then we can derive P, Q, and B, as has also been indicated before.

3.8.3 Subspace Parameter Estimation—Implementation Aspects

Let $\hat{R}_{fp}$ be the sample estimate

$$\hat{R}_{fp} = \frac{1}{N} \sum_{t=m+1}^{N-m+1} f(t)p^*(t) \tag{3.8.42}$$

and let $\hat{R}_{ff}$ *etc.* be similarly defined. Compute $\hat{R}$ as

$$\hat{R} = \hat{R}_{ff}^{-1/2} \hat{R}_{fp} \hat{R}_{pp}^{-1/2} \tag{3.8.43}$$

and its SVD. Estimate n as the "practical rank" of $\hat{R}$, or

$$\boxed{\hat{n} = \text{p-rank}(\hat{R})} \tag{3.8.44}$$

(i.e., the number of singular values of $\hat{R}$ that are significantly larger than the remaining ones; statistical tests for deciding whether a singular value of a given sample covariance matrix is significantly different from zero are discussed in, for example, [Fuchs 1987]). Let $\hat{U}$, $\hat{\Sigma}$, and $\hat{V}$ denote the matrices made from the $\hat{n}$ principal singular elements of $\hat{R}$, corresponding to the matrices U, Σ, and V in (3.8.39). Take

$$\boxed{\begin{aligned} \hat{C} &= \text{the first } ny \text{ rows of } \hat{R}_{ff}^{1/2} \hat{U} \hat{\Sigma}^{1/2} \\ \hat{D} &= \text{the first } ny \text{ columns of } \hat{\Sigma}^{1/2} \hat{V}^* \hat{R}_{pp}^{1/2} \end{aligned}} \tag{3.8.45}$$

Next, let

$$\begin{aligned} \bar{\Gamma} \text{ and } \underline{\Gamma} = &\text{ the matrices made from the first and, respectively, last} \\ &(m-1) \text{ block rows of } \hat{R}_{ff}^{1/2} \hat{U} \hat{\Sigma}^{1/2} \end{aligned} \tag{3.8.46}$$

Estimate A as

$$\boxed{\hat{A} = \text{the LS or TLS solution to } \bar{\Gamma} A \simeq \underline{\Gamma}} \tag{3.8.47}$$

Finally, estimate P as

$$\boxed{\hat{P} = \text{the positive definite solution, if any, of the Riccati equation (3.8.19) with } A,\ C,\ D, \text{ and } R_0 \text{ replaced by their estimates}} \tag{3.8.48}$$

and estimate Q and B

$$\boxed{\begin{aligned} \hat{Q} &= \hat{R}_0 - \hat{C}\hat{P}\hat{C}^* \\ \hat{B} &= (\hat{D} - \hat{A}\hat{P}\hat{C}^*)\hat{Q}^{-1} \end{aligned}} \tag{3.8.49}$$

In some cases, the procedure cannot be completed, because the Riccati equation has no positive definite solution or even no solution at all. (In the case of a real-valued ARMA signal, for instance, that equation could have no real-valued solution.) In such cases, we can estimate *approximate P* as discussed next (only the estimation of P has to be modified; all the other parameter estimates can be obtained as previously described).

A straightforward calculation making use of (3.8.11) and (3.8.12) yields

$$\begin{aligned} E\left\{x(t)y^*(t-k)\right\} &= A^k PC^* + A^{k-1}BQ \\ &= A^{k-1}D \qquad (\text{for } k \geq 1) \end{aligned} \tag{3.8.50}$$

Hence,

$$\mathcal{C}^* = E\left\{x(t)p^*(t)\right\} \tag{3.8.51}$$

Let

$$\psi = \mathcal{C}^* R_{pp}^{-1} \tag{3.8.52}$$

and define $\epsilon(t)$ via the equation

$$x(t) = \psi p(t) + \epsilon(t) \tag{3.8.53}$$

It is not difficult to verify that $\epsilon(t)$ is uncorrelated with $p(t)$. Indeed,

$$E\left\{\epsilon(t)p^*(t)\right\} = E\left\{[x(t) - \psi p(t)]p^*(t)\right\} = \mathcal{C}^* - \psi R_{pp} = 0 \tag{3.8.54}$$

This implies that *the first term in (3.8.53) is the least-squares approximation of $x(t)$ based on the past signal values in $p(t)$*. (See, for example, [Söderström and Stoica 1989] and Appendix A.) It follows from this observation that $\psi p(t)$ approaches $x(t)$ as m increases. Hence,

$$\psi R_{pp}\psi^* = \mathcal{C}^* R_{pp}^{-1}\mathcal{C} \to P \qquad (\text{as } m \to \infty) \tag{3.8.55}$$

However, in view of (3.8.41),

$$\mathcal{C}^* R_{pp}^{-1}\mathcal{C} = \Sigma \tag{3.8.56}$$

The conclusion is that, provided m is chosen large enough, we can approximate P as

$$\boxed{\tilde{P} = \hat{\Sigma}, \qquad \text{for } m \gg 1} \tag{3.8.57}$$

This is the alternative estimate of P, which can be used in lieu of (3.8.48) whenever the latter estimation procedure fails. The estimate $\tilde{P}$ approaches the true value P as N tends to infinity, *provided* m is also increased without bound at an appropriate rate. However, if (3.8.57) is used with too small a value of m, the estimate of P so obtained might be heavily biased.

The reader interested in more aspects of the subspace approach to parameter estimation for rational models should consult [Aoki 1987; van Overschee and de Moor 1996; Rao and Arun 1992; Viberg 1995] and the references therein.

3.9 COMPLEMENTS

3.9.1 The Partial Autocorrelation Sequence

The sequence $\{k_j\}$ computed in equation (3.5.7) of the LDA has an interesting statistical interpretation, as explained next. The covariance lag ρ_j "measures" the degree of correlation between the data samples $y(t)$ and $y(t-j)$ (in the chapter ρ_j is equal to either $r(j)$ or $\hat{r}(j)$; here $\rho_j = r(j)$). The normalized covariance sequence $\{\rho_j/\rho_0\}$ is often called the *autocorrelation function*. Now, $y(t)$ and $y(t-j)$ are related to one another not only "directly," but also through the intermediate samples:

$$[y(t-1)\dots y(t-j+1)]^T \triangleq \varphi(t)$$

Let $\epsilon_f(t)$ and $\epsilon_b(t-j)$ denote the errors of the LS linear predictions of $y(t)$ and $y(t-j)$, respectively, based on $\varphi(t)$ above; in particular, $\epsilon_f(t)$ and $\epsilon_b(t-j)$ must then be uncorrelated with $\varphi(t)$: $E\left\{\epsilon_f(t)\varphi^*(t)\right\} = E\left\{\epsilon_b(t-j)\varphi^*(t)\right\} = 0$. (Note that $\epsilon_f(t)$ and $\epsilon_b(t-j)$ are termed forward and backward prediction errors respectively; see also Exercises 3.3 and 3.4.) We show that

$$\boxed{k_j = -\frac{E\left\{\epsilon_f(t)\epsilon_b^*(t-j)\right\}}{\left[E\left\{|\epsilon_f(t)|^2\right\} E\left\{|\epsilon_b(t-j)|^2\right\}\right]^{1/2}}} \tag{3.9.1}$$

Hence, k_j is the negative of the so-called *partial correlation* (PARCOR) coefficient of $\{y(t)\}$, which measures the "partial correlation" between $y(t)$ and $y(t-j)$ after the correlation due to the intermediate values $y(t-1), \dots, y(t-j+1)$ has been eliminated.

Let

$$\epsilon_f(t) = y(t) + \varphi^T(t)\theta \tag{3.9.2}$$

where, similarly to (3.4.9),

$$\theta = -\{E\left\{\varphi^c(t)\varphi^T(t)\right\}\}^{-1}\, E\left\{\varphi^c(t)\, y(t)\right\} \triangleq -R^{-1}r$$

It is readily verified (by making use of the previous definition for θ) that

$$E\left\{\varphi^c(t)\epsilon_f(t)\right\} = 0$$

which shows that $\epsilon_f(t)$, as just defined, is indeed the error of the linear *forward* LS prediction of $y(t)$, based on $\varphi(t)$.

Similarly, define the linear *backward* LS prediction error

$$\epsilon_b(t-j) = y(t-j) + \varphi^T(t)\alpha$$

where

$$\alpha = -\{E\left\{\varphi^c(t)\varphi^T(t)\right\}\}^{-1}E\left\{\varphi^c(t)y(t-j)\right\} = -R^{-1}\tilde{r} = \tilde{\theta}$$

The last equality just defined follows from (3.5.3). We thus have

$$E\left\{\varphi^c(t)\epsilon_b(t-j)\right\} = 0$$

as required.

Next, some simple calculations give

$$\begin{aligned} E\left\{|\epsilon_f(t)|^2\right\} &= E\left\{y^*(t)[y(t) + \varphi^T(t)\theta]\right\} \\ &= \rho_0 + [\rho_1^* \dots \rho_{j-1}^*]\theta = \sigma_{j-1}^2 \end{aligned}$$

$$\begin{aligned} E\left\{|\epsilon_b(t-j)|^2\right\} &= E\left\{y^*(t-j)[y(t-j) + \varphi^T(t)\alpha]\right\} \\ &= \rho_0 + [\rho_{j-1} \dots \rho_1]\tilde{\theta} = \sigma_{j-1}^2 \end{aligned}$$

and

$$\begin{aligned} E\left\{\epsilon_f(t)\epsilon_b^*(t-j)\right\} &= E\left\{[y(t) + \varphi^T(t)\theta]y^*(t-j)\right\} \\ &= \rho_j + [\rho_{j-1} \dots \rho_1]\theta = \alpha_{j-1} \end{aligned}$$

(*cf.* (3.4.1) and (3.5.6)). By using the previous equations in (3.9.1), we obtain

$$k_j = -\alpha_{j-1}/\sigma_{j-1}^2$$

which coincides with (3.5.7).

3.9.2 Some Properties of Covariance Extensions

Assume we are given a finite sequence $\{r(k)\}_{k=-(m-1)}^{m-1}$ with $r(-k) = r^*(k)$ and such that R_m in equation (3.4.6) is positive definite. We show that the finite sequence can be extended to an infinite sequence that is a valid ACS. Moreover, there are an infinite number of possible covariance extensions, and we derive an algorithm to construct these extensions. One such extension, in which the reflection coefficients $k_m, k_{m+1}, \dots$ are all zero (and thus the infinite ACS corresponds to an AR process of order less than or equal to $(m-1)$), gives the so-called Maximum Entropy extension [Burg 1975].

We begin by constructing the set of $r(m)$ values for which $R_{m+1} > 0$. Using the result of Exercise 3.7, we have

$$|R_{m+1}| = \sigma_m^2 |R_m| \tag{3.9.3}$$

From the Levinson–Durbin algorithm,

$$\sigma_m^2 = \sigma_{m-1}^2 \left[1 - |k_m|^2\right] = \sigma_{m-1}^2 \left[1 - \frac{|r(m) + \tilde{r}_{m-1}^* \theta_{m-1}|^2}{\sigma_{m-1}^4}\right] \tag{3.9.4}$$

Combining (3.9.3) and (3.9.4) gives

$$|R_{m+1}| = |R_m| \cdot \sigma_{m-1}^2 \left[1 - \frac{|r(m) + \tilde{r}_{m-1}^* \theta_{m-1}|^2}{\sigma_{m-1}^4}\right] \tag{3.9.5}$$

which shows that $|R_{m+1}|$ is quadratic in $r(m)$. Since $\sigma_{m-1}^2 > 0$ and R_m is positive definite, it follows that

$$|R_{m+1}| > 0 \text{ if and only if } |r(m) + \tilde{r}_{m-1}^* \theta_{m-1}|^2 < \sigma_{m-1}^4 \tag{3.9.6}$$

This region is an open disk in the complex plane whose center is $-\tilde{r}_{m-1}^* \theta_{m-1}$ and radius is σ_{m-1}^2.

Equation (3.9.6) leads to a construction of all possible covariance extensions. Note that, if $R_p > 0$ and we choose $r(p)$ inside the disk $|r(p) + \tilde{r}_{p-1}^* \theta_{p-1}|^2 < \sigma_{p-1}^4$, then $|R_{p+1}| > 0$. This implies $\sigma_p^2 > 0$, and the admissible disk for $r(p+1)$ has nonzero radius, so there are an infinite number of possible choices for $r(p+1)$ such that $|R_{p+2}| > 0$. Arguing inductively in this way for $p = m, m+1, \ldots$ shows that there are an infinite number of covariance extensions and provides a construction for them.

If we choose $r(p) = -\tilde{r}_{p-1}^* \theta_{p-1}$ for $p = m, m+1, \ldots$ (i.e., $r(p)$ is chosen to be at the center of each disk in (3.9.6)), then, from (3.9.4), we see that the reflection coefficient $k_p = 0$. Thus, from the Levinson–Durbin algorithm (see equation (3.5.10)) we have

$$\theta_p = \begin{bmatrix} \theta_{p-1} \\ 0 \end{bmatrix} \tag{3.9.7}$$

and

$$\sigma_p^2 = \sigma_{p-1}^2 \tag{3.9.8}$$

Arguing inductively again, we find that $k_p = 0$, $\theta_p = \begin{bmatrix} \theta_{m-1} \\ 0 \end{bmatrix}$, and $\sigma_p^2 = \sigma_{m-1}^2$ for $p = m, m+1, \ldots$. This extension, called the Maximum Entropy extension [Burg 1975], thus gives an ACS sequence that corresponds to an AR process of order less than or equal to $(m-1)$. The name *maximum entropy* arises because the spectrum so obtained has maximum entropy rate $\int_{-\pi}^{\pi} \ln \phi(\omega) d\omega$ under the Gaussian assumption [Burg 1975]; the entropy rate is closely related to the numerator in the spectral-flatness measure introduced in Exercise 3.6.

For some recent results on the covariance-extension problem and its variations, we refer to [Byrnes, Georgiou, and Lindquist 2001] and the references therein.

3.9.3 The Burg Method for AR Parameter Estimation

The thesis [Burg 1975] developed a method for AR parameter estimation that is based on forward and backward prediction errors and on direct estimation of the reflection coefficients in equation (3.9.1). In this complement, we develop the Burg estimator and discuss some of its properties.

Assume we have data measurements $\{y(t)\}$ for $t = 1, 2, \ldots, N$. Much as in Complement 3.9.1, we define the forward and backward prediction errors for a pth-order model as

$$\hat{e}_{f,p}(t) = y(t) + \sum_{i=1}^{p} \hat{a}_{p,i} y(t-i), \qquad t = p+1, \ldots, N \tag{3.9.9}$$

$$\hat{e}_{b,p}(t) = y(t-p) + \sum_{i=1}^{p} \hat{a}^*_{p,i} y(t-p+i), \qquad t = p+1, \ldots, N \tag{3.9.10}$$

We have shifted the time index in the definition of $e_b(t)$ from that in equation (3.9.2) to reflect that $\hat{e}_{b,p}(t)$ is computed from data up to time t; also, the fact that the coefficients in (3.9.10) are given by $\{\hat{a}^*_{p,i}\}$ follows from Complement 3.9.1. We use hats to denote estimated quantities, and we explicitly denote the order p in both the prediction error sequences and the AR coefficients. The AR parameters are related to the reflection coefficient $\hat{k}_p$ by (see (3.5.10))

$$\hat{a}_{p,i} = \begin{cases} \hat{a}_{p-1,i} + \hat{k}_p \hat{a}^*_{p-1,p-i}, & i = 1, \ldots, p-1 \\ \hat{k}_p, & i = p \end{cases} \tag{3.9.11}$$

Burg's method considers the recursive-in-order estimation of $\hat{k}_p$ *given that the AR coefficients for order* $p-1$ *have been computed.* In particular, Burg's method finds $\hat{k}_p$ to minimize the arithmetic mean of the forward and backward prediction-error variance estimates, namely,

$$\min_{\hat{k}_p} \frac{1}{2} \left[\hat{\rho}_f(p) + \hat{\rho}_b(p) \right] \tag{3.9.12}$$

where

$$\hat{\rho}_f(p) = \frac{1}{N-p} \sum_{t=p+1}^{N} \left| \hat{e}_{f,p}(t) \right|^2$$

$$\hat{\rho}_b(p) = \frac{1}{N-p} \sum_{t=p+1}^{N} \left| \hat{e}_{b,p}(t) \right|^2$$

and where $\{\hat{a}_{p-1,i}\}_{i=1}^{p-1}$ are assumed to be known from the recursion at the previous order.

The prediction errors satisfy the following recursive-in-order expressions:

$$\hat{e}_{f,p}(t) = \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1) \tag{3.9.13}$$

$$\hat{e}_{b,p}(t) = \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t) \tag{3.9.14}$$

Equation (3.9.13) follows directly from (3.9.9)–(3.9.11) as

$$\begin{aligned}
\hat{e}_{f,p}(t) &= y(t) + \sum_{i=1}^{p-1} \left(\hat{a}_{p-1,i} + \hat{k}_p \hat{a}_{p-1,p-i}^* \right) y(t-i) + \hat{k}_p y(t-p) \\
&= \left[y(t) + \sum_{i=1}^{p-1} \hat{a}_{p-1,i} y(t-i) \right] + \hat{k}_p \left[y(t-p) + \sum_{i=1}^{p-1} \hat{a}_{p-1,i}^* y(t-p+i) \right] \\
&= \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\hat{e}_{b,p}(t) &= y(t-p) + \sum_{i=1}^{p-1} [\hat{a}_{p-1,i}^* + \hat{k}_p^* \hat{a}_{p-1,p-i}] y(t-p+i) + \hat{k}_p^* y(t) \\
&= \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t)
\end{aligned}$$

which shows (3.9.14).

We can use the previous expressions to develop a recursive-in-order algorithm for estimating the AR coefficients. Note that the quantity to be minimized in (3.9.12) is quadratic in $\hat{k}_p$, because

$$\begin{aligned}
\frac{1}{2}\left[\hat{\rho}_f(p) + \hat{\rho}_b(p)\right] &= \frac{1}{2(N-p)} \sum_{t=p+1}^{N} \left\{ \left| \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1) \right|^2 \right. \\
&\qquad \left. + \left| \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t) \right|^2 \right\} \\
&= \frac{1}{2(N-p)} \sum_{t=p+1}^{N} \left\{ \left[\left| \hat{e}_{f,p-1}(t) \right|^2 + \left| \hat{e}_{b,p-1}(t-1) \right|^2 \right] \left[1 + |\hat{k}_p|^2 \right] \right. \\
&\qquad + 2\hat{e}_{f,p-1}(t) \hat{e}_{b,p-1}^*(t-1) \hat{k}_p^* \\
&\qquad \left. + 2\hat{e}_{f,p-1}^*(t) \hat{e}_{b,p-1}(t-1) \hat{k}_p \right\}
\end{aligned}$$

Using Result R34 in Appendix A, we find that the $\hat{k}_p$ that minimizes the above quantity is given by

$$\hat{k}_p = \frac{-2 \sum_{t=p+1}^{N} \hat{e}_{f,p-1}(t) \hat{e}_{b,p-1}^*(t-1)}{\sum_{t=p+1}^{N} \left[\left| \hat{e}_{f,p-1}(t) \right|^2 + \left| \hat{e}_{b,p-1}(t-1) \right|^2 \right]} \tag{3.9.15}$$

A recursive-in-order algorithm for estimating the AR parameters, called the *Burg algorithm*, is as follows:

The Burg Algorithm

Step 0 Initialize $\hat{e}_{f,0}(t) = \hat{e}_{b,0}(t) = y(t)$.

Step 1 For $p = 1, \ldots, n$,

(a) Compute $\hat{e}_{f,p-1}(t)$ and $\hat{e}_{b,p-1}(t)$ for $t = p+1, \ldots, N$ from (3.9.13) and (3.9.14).

(b) Compute $\hat{k}_p$ from (3.9.15).

(c) Compute $\hat{a}_{p,i}$ for $i = 1, \ldots, p$ from (3.9.11).

Then $\hat{\theta} = [\hat{a}_{p,1}, \ldots, \hat{a}_{p,p}]^T$ is the vector of AR coefficient estimates.

Finally, we show that the resulting AR model is stable; this is accomplished by showing that $|\hat{k}_p| \leq 1$ for $p = 1, \ldots, n$. (See Exercise 3.9.) To do so, we express $\hat{k}_p$ as

$$\hat{k}_p = \frac{-2c^*d}{c^*c + d^*d} \tag{3.9.16}$$

where

$$c = [\hat{e}_{b,p-1}(p), \ldots, \hat{e}_{b,p-1}(N-1)]^T$$
$$d = [\hat{e}_{f,p-1}(p+1), \ldots, \hat{e}_{f,p-1}(N)]^T$$

Then

$$0 \leq \|c - e^{i\alpha}d\|^2 = c^*c + d^*d - 2\,\mathrm{Re}\,\{e^{i\alpha}c^*d\} \quad \text{for every } \alpha \in [-\pi, \pi]$$
$$\Longrightarrow\ 2\,\mathrm{Re}\,\{e^{i\alpha}c^*d\} \leq c^*c + d^*d \quad \text{for every } \alpha \in [-\pi, \pi]$$
$$\Longrightarrow\ 2|c^*d| \leq c^*c + d^*d \ \Longrightarrow\ |\hat{k}_p| \leq 1$$

The Burg algorithm is computationally simple, and it is amenable to both order-recursive and time-recursive solutions. In addition, the Burg AR model estimate is guaranteed to be stable. On the other hand, the Burg method is suboptimal, in that it estimates the n reflection coefficients by decoupling an n-dimensional minimization problem into the n one-dimensional minimizations in (3.9.12). This is in contrast to the LS AR method in Section 3.4.2, in which the AR coefficients are found by an n-dimensional minimization. For large N, the two algorithms give very similar performance; for short or medium data lengths, the Burg algorithm usually behaves somewhere between the LS method and the Yule–Walker method.

3.9.4 The Gohberg–Semencul Formula

The Hermitian Toeplitz matrix R_{n+1} in (3.4.6) is highly structured. In particular, it is completely defined by its first column (or row). As was shown in Section 3.5, exploitation of the special algebraic structure of (3.4.6) makes it possible to solve this system of equations very efficiently. In this complement, we show that the Toeplitz structure of R_{n+1} may also be exploited to derive a *closed-form* expression for the inverse of this matrix. This expression is what is usually called the *Gohberg–Semencul (GS) formula* (or the Gohberg–Semencul–Heining formula, in recognition of the contribution also made by Heining to its discovery) [Söderström and Stoica 1989; Iohvidov 1982; Böttcher and Silbermann 1983]. As will be seen, an interesting consequence of the GS formula is the fact that, even if R_{n+1}^{-1} is *not* Toeplitz in general, it is still completely determined by its first column. Observe from (3.4.6) that the first column of R_{n+1}^{-1} is given by $[1 \;\; \theta]^T/\sigma^2$. In what follows, we drop the subscript n of θ for notational convenience.

The derivation of the GS formula requires some preparations. First, note that the following nested structures of R_{n+1},

$$R_{n+1} = \begin{bmatrix} \rho_0 & r_n^* \\ r_n & R_n \end{bmatrix} = \begin{bmatrix} R_n & \tilde{r}_n \\ \tilde{r}_n^* & \rho_0 \end{bmatrix}$$

along with (3.4.6) and the result (3.5.3), imply that

$$\theta = -R_n^{-1} r_n, \qquad \tilde{\theta} = -R_n^{-1} \tilde{r}_n$$

$$\sigma_n^2 = \rho_0 - r_n^* R_n^{-1} r_n = \rho_0 - \tilde{r}_n^* R_n^{-1} \tilde{r}_n$$

Next, make use of the above equations and a standard formula for the inverse of a partitioned matrix (see Result R26 in Appendix A) to write

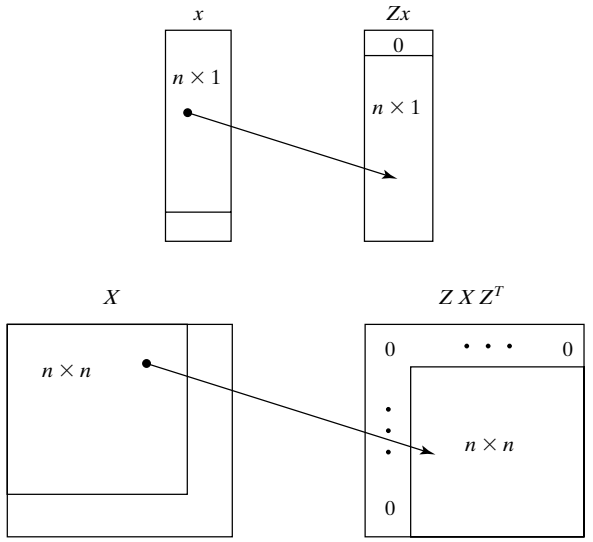
$$R_{n+1}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & R_n^{-1} \end{bmatrix} + \begin{bmatrix} 1 \\ \theta \end{bmatrix} [1 \;\; \theta^*]/\sigma_n^2 \tag{3.9.17}$$

$$= \begin{bmatrix} R_n^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \tilde{\theta} \\ 1 \end{bmatrix} [\tilde{\theta}^* \;\; 1]/\sigma_n^2 \tag{3.9.18}$$

Finally, introduce the $(n+1) \times (n+1)$ matrix

$$Z = \begin{bmatrix} 0 & \dots & 0 \\ 1 & \ddots & \vdots \\ & \ddots & \\ 0 & & 1 \;\; 0 \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ & & \vdots \\ I_{n \times n} & & \\ & & 0 \end{bmatrix}$$

and observe that multiplication by Z of a vector or a matrix has the effects indicated here:



Owing to these effects of the linear transformation by Z, this matrix is called a *shift* or *displacement operator*.

We are now prepared to present a simple derivation of the GS formula. The basic idea of this derivation is to eliminate R_n^{-1} from the expressions for R_{n+1}^{-1} in (3.9.17) and (3.9.18) by making use of the displacement properties of Z. Hence, using the expression (3.9.17) for R_{n+1}^{-1}, and its "dual" (3.9.18) for calculating $ZR_{n+1}^{-1}Z^T$, gives

$$R_{n+1}^{-1} - ZR_{n+1}^{-1}Z^T = \frac{1}{\sigma_n^2}\left\{\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}[1 \quad a_1^* \ldots a_n^*] - \begin{bmatrix} 0 \\ a_n^* \\ \vdots \\ a_1^* \end{bmatrix}[0 \quad a_n \ldots a_1]\right\} \tag{3.9.19}$$

Premultiplying and postmultiplying (3.9.19) by Z and Z^T, respectively, and then continuing to do so with the resulting equations, we obtain

$$ZR_{n+1}^{-1}Z^T - Z^2R_{n+1}^{-1}Z^{2T} = \frac{1}{\sigma_n^2}\left\{\begin{bmatrix} 0 \\ 1 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}[0 \quad 1 \quad a_1^* \ldots a_{n-1}^*] - \begin{bmatrix} 0 \\ 0 \\ a_n^* \\ \vdots \\ a_2^* \end{bmatrix}[0 \quad 0 \quad a_n \ldots a_2]\right\} \tag{3.9.20}$$

$$\vdots$$

$$Z^n R_{n+1}^{-1} Z^{nT} - 0 = \frac{1}{\sigma_n^2} \left\{ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} [0 \dots 0 \quad 1] \right\} \tag{3.9.21}$$

In (3.9.21), use is made of the fact that Z is a *nilpotent matrix of order* $n+1$, in the sense that

$$Z^{n+1} = 0$$

(which can be readily verified). Now, by simply summing up equations (3.9.19)–(3.9.21), we derive the following expression for R_{n+1}^{-1}:

$$\begin{aligned} R_{n+1}^{-1} = \frac{1}{\sigma_n^2} \left\{ \begin{bmatrix} 1 & & 0 \\ a_1 & \ddots & \\ \vdots & \ddots & \ddots \\ a_n & \dots & a_1 \quad 1 \end{bmatrix} \begin{bmatrix} 1 & a_1^* & \dots & a_n^* \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_1^* \\ 0 & & & 1 \end{bmatrix} \right. \\ \left. - \begin{bmatrix} 0 & & 0 \\ a_n^* & \ddots & \\ \vdots & \ddots & \ddots \\ a_1^* & \dots & a_n^* \quad 0 \end{bmatrix} \begin{bmatrix} 0 & a_n & \dots & a_1 \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_n \\ 0 & & & 0 \end{bmatrix} \right\} \end{aligned} \tag{3.9.22}$$

This is the GS formula. Note from (3.9.22) that R_{n+1}^{-1} is, indeed, completely determined by its first column, as claimed earlier.

The GS formula is inherently related to the Yule–Walker method of AR modeling, and this is one of the reasons for including it in this book. The GS formula is also useful in studying other spectral estimators, such as the Capon method, which is discussed in Chapter 5. The hope that the curious reader who studies this part will become interested in the fascinating topic of Toeplitz matrices and allied subjects is another reason for its inclusion. In particular, it is indeed fascinating to be able to derive an analytical formula for the inverse of a given matrix, as has been shown to be the case for Toeplitz matrices. The basic ideas of the previous derivation may be extended to more general matrices. Let us explain this briefly. For a given matrix X, the rank of $X - ZXZ^T$ is called the *displacement rank* of X under Z. As can be seen from (3.9.19), the inverse of a Hermitian Toeplitz matrix has a displacement rank equal to two. Now, assume we are given a (structured) matrix X for which we are able to find a nilpotent matrix Y such that X^{-1} has a *low* displacement rank under Y; the matrix Y does not need to have the previous form of Z. Then, paralleling the calculations in (3.9.19)–(3.9.22), we might be able to derive a simple "closed-form" expression for X^{-1}. See [Friedlander, Morf, Kailath, and Ljung 1979] for more details on the topic of this complement.

3.9.5 MA Parameter Estimation in Polynomial Time

The parameter estimation of an AR process via the LS method leads to a quadratic minimization problem that can be solved in closed form (see (3.4.11), (3.4.12)). On the other hand, for an MA process, the LS criterion similar to (3.4.11), which is given by

$$\sum_{t=N_1}^{N_2} \left| \frac{1}{B(z)} y(t) \right|^2 \tag{3.9.23}$$

is a highly nonlinear function of the MA parameters (and likewise for an ARMA process).

A simple MA spectral estimator, one that does not require solving a nonlinear minimization problem, is given by equation (3.6.4) and is repeated here:

$$\hat{\phi}(\omega) = \sum_{k=-\hat{m}}^{\hat{m}} \hat{r}(k) e^{-i\omega k} \tag{3.9.24}$$

where $\hat{m}$ is the assumed MA order and $\{\hat{r}(k)\}$ are the standard sample covariances. As explained in Section 3.6, the main problem associated with (3.9.24) is the fact that $\hat{\phi}(\omega)$ is not guaranteed to be positive for all $\omega \in [0, 2\pi]$. If the final goal of the signal processing exercise is spectral analysis, then an occurrence of negative values $\hat{\phi}(\omega) < 0$ (for some values of ω) is not acceptable, as the true spectral density of course satisfies $\phi(\omega) \geq 0$ for all $\omega \in [0, 2\pi]$. If the goal is MA parameter estimation, then the problem induced by $\hat{\phi}(\omega) < 0$ (for some values of ω) is even more serious, because, in such a case, $\hat{\phi}(\omega)$ cannot be factored as in (3.6.1), and hence, *no* MA parameter estimates can be obtained directly from $\hat{\phi}(\omega)$. In this complement, we will show how to get around the problem of $\hat{\phi}(\omega) < 0$ and, hence, how to obtain MA parameter estimates from such an invalid MA spectral density estimate, using an indirect but computationally efficient method. (See [Stoica, McKelvey, and Mari 2000; Dumitrescu, Tabus, and Stoica 2001].) Note that obtaining MA parameter estimates from the $\hat{\phi}(\omega)$ in (3.9.24) is of interest not only for MA estimation, but also as a step of some ARMA estimation methods. (See, for example, (3.7.9) as well as Exercise 3.12.)

A sound way of tackling this problem of "factoring the unfactorable" is as follows: Let $\phi(\omega)$ denote the PSD of an MA process of order m; that is,

$$\phi(\omega) = \sum_{k=-m}^{m} r(k) e^{-i\omega k} \geq 0, \quad \omega \in [0, 2\pi] \tag{3.9.25}$$

We would like to find the $\phi(\omega)$ in (3.9.25) that is closest to $\hat{\phi}(\omega)$ in (3.9.24), in the following LS sense:

$$\min \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\hat{\phi}(\omega) - \phi(\omega) \right]^2 d\omega \tag{3.9.26}$$

The order m in (3.9.25) could be different from the order $\hat{m}$ in (3.9.24). Without loss of generality, we can assume that $m \le \hat{m}$. Indeed, if $m > \hat{m}$, we can extend the sequence $\{\hat{r}(k)\}$ with zeroes to make $m \le \hat{m}$. Once $\phi(\omega)$ has been obtained by solving (3.9.26), we can factor it by using any of a number of available *spectral factorization algorithms* (see, for example, [Wilson 1969; Vostry 1975; Vostry 1976]) and, in this way, derive MA parameter estimates $\{b_k\}$ satisfying

$$\phi(\omega) = \sigma^2 |B(\omega)|^2 \tag{3.9.27}$$

(See (3.6.1).) This step for obtaining $\{b_k\}$ and σ^2 from $\phi(\omega)$ can be computed in $\mathcal{O}(m^2)$ flops. The problem that remains is to solve (3.9.26) for $\phi(\omega)$ in a similar number of flops.

Now,

$$\hat{\phi}(\omega) - \phi(\omega) = \sum_{k=-m}^{m} \left[\hat{r}(k) - r(k)\right] e^{-i\omega k} + \sum_{|k|>m} \hat{r}(k) e^{-i\omega k}$$

so it follows from Parseval's theorem (see (1.2.6)) that the spectral LS criterion of (3.9.26) can be rewritten as a covariance fitting criterion:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\hat{\phi}(\omega) - \phi(\omega)\right]^2 d\omega = \sum_{k=-m}^{m} \left|\hat{r}(k) - r(k)\right|^2 + \sum_{|k|>m} \left|\hat{r}(k)\right|^2$$

Consequently, the approximation problem (3.9.26) is equivalent to

$$\min_{\{r(k)\}} \|\hat{r} - r\|_W^2 \text{ subject to (3.9.25)} \tag{3.9.28}$$

where $\|x\|_W^2 = x^* W x$ and

$$\hat{r} = \begin{bmatrix} \hat{r}(0) & \dots & \hat{r}(m) \end{bmatrix}^T$$

$$r = \begin{bmatrix} r(0) & \dots & r(m) \end{bmatrix}^T$$

$$W = \begin{bmatrix} 1 & & & 0 \\ & 2 & & \\ & & \ddots & \\ 0 & & & 2 \end{bmatrix}$$

Next, we will describe a computationally efficient and reliable algorithm for solving problem (3.9.28) (with a *general* W matrix) in a time that is a polynomial function of m (a more precise flop count is given below). A possible way of tackling (3.9.28) would be to first write the covariances $\{r(k)\}$ as functions of the MA parameters (see (3.3.3)), which would guarantee that they satisfy (3.9.25), and to then minimize the function in (3.9.28) with respect to the MA parameters. However, the minimization problem so obtained would, much like (3.9.23), be nonlinear in the MA parameters (more precisely, the criterion in (3.9.28) is *quartic* in $\{b_k\}$), which is exactly the type of problem we tried to avoid in the first place.

As a preparation step for solving (3.9.28), we first derive a parameterization of the MA covariance sequence $\{r(k)\}$, which will turn out to be more convenient than the parameterization via $\{b_k\}$. Let J_k denote the $(m+1)\times(m+1)$ matrix with ones on the $(k+1)$st diagonal and zeroes everywhere else:

$$J_k = \begin{bmatrix} 0 & \cdots & 0 & \overbrace{1}^{k+1} & & 0 \\ \vdots & & & 0 & \ddots & \\ & & & & \ddots & 1 \\ \vdots & & 0 & & & 0 \\ & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & & 0 \end{bmatrix}, \quad (m+1)\times(m+1)$$

(for $k = 0, \ldots, m$). Note that $J_0 = I$. Then the following result holds:

> Any MA covariance sequence $\{r(k)\}_{k=0}^{m}$ can be written as $r(k) = \operatorname{tr}(J_k Q)$ for $k = 0, \ldots, m$, where Q is an $(m+1)\times(m+1)$ positive semidefinite matrix. (3.9.29)

To prove this result, let

$$a(\omega) = \begin{bmatrix} 1 & e^{i\omega} & \cdots & e^{im\omega} \end{bmatrix}^T$$

and observe that

$$a(\omega)a^*(\omega) = \begin{bmatrix} 1 & e^{-i\omega} & \cdots & e^{-im\omega} \\ e^{i\omega} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & e^{-i\omega} \\ e^{im\omega} & \cdots & e^{i\omega} & 1 \end{bmatrix} = \sum_{k=-m}^{m} J_k e^{-ik\omega}$$

where $J_{-k} = J_k^T$ (for $k \geq 0$). Hence, for the sequence parameterized as in (3.9.29), we have that

$$\begin{aligned} \sum_{k=-m}^{m} r(k)e^{-ik\omega} &= \operatorname{tr}\left[\sum_{k=-m}^{m} J_k Q e^{-ik\omega}\right] \\ &= \operatorname{tr}\left[a(\omega)a^*(\omega)Q\right] = a^*(\omega)Qa(\omega) \geq 0, \quad \text{for } \omega \in [0, 2\pi] \end{aligned}$$

which implies that $\{r(k)\}$ indeed is an MA(m) covariance sequence. To show that any MA(m) covariance sequence can be parameterized as in (3.9.29), we make use of (3.3.3) to write

(for $k = 0, \ldots, m$)

$$r(k) = \sigma^2 \sum_{j=k}^{m} b_j b_{j-k}^* = \sigma^2 \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix} J_k \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix}$$

$$= \operatorname{tr} \left\{ J_k \cdot \sigma^2 \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix} \right\} \tag{3.9.30}$$

Evidently (3.9.30) has the form stated in (3.9.29) with

$$Q = \sigma^2 \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix}$$

With this observation, the proof of (3.9.29) is complete.

We can now turn our attention to the main problem, (3.9.28). We will describe an efficient algorithm for solving (3.9.28) with a general weighting matrix $W > 0$ (as already stated.). For a choice of W that usually yields more accurate MA parameter estimates than the simple diagonal weighting in (3.9.28), we refer the reader to [Stoica, McKelvey, and Mari 2000]. Let

$$\mu = C(\hat{r} - r)$$

where C is the Cholesky factor of W (i.e., C is an upper triangular matrix and $W = C^*C$). Also, let α be a vector containing all the elements in the upper triangle of Q, including the diagonal:

$$\alpha = \left[Q_{1,1} \; Q_{1,2} \; \cdots \; Q_{1,m+1} \; ; \; Q_{2,2} \; \cdots \; Q_{2,m+1} \; ; \cdots \; ; Q_{m+1,m+1} \right]^T$$

Note that α defines Q; that is, the elements of Q are either elements of α or complex conjugates of elements of α. Making use of this notation and of (3.9.29), we can rewrite (3.9.28) in the following form (for real-valued sequences):

$$\begin{aligned} &\min_{\rho,\mu,\alpha} \rho \quad \text{subject to:} \\ &\|\mu\| \le \rho \\ &Q \ge 0 \\ &\begin{bmatrix} \operatorname{tr}[Q] \\ \operatorname{tr}\left[\frac{1}{2}\left(J_1 + J_1^T\right) Q\right] \\ \vdots \\ \operatorname{tr}\left[\frac{1}{2}\left(J_m + J_m^T\right) Q\right] \end{bmatrix} + C^{-1}\mu = \hat{r} \end{aligned} \tag{3.9.31}$$

Note that, to obtain the equality constraint in (3.9.31), we used the fact that (in the *real-valued case*; the complex-valued case can be treated similarly)

$$r(k) = \mathrm{tr}(J_k Q) = \mathrm{tr}(Q^T J_k^T) = \mathrm{tr}(J_k^T Q) = \frac{1}{2}\,\mathrm{tr}\left[(J_k + J_k^T)Q\right]$$

The reason for this seemingly artificial trick is that we need the matrices multiplying Q in (3.9.31) to be symmetric. In effect, the problem (3.9.31) has precisely the form of a *semidefinite quadratic program* (SQP), which can be solved efficiently by means of interior-point methods (see [Sturm 1999] and also [Dumitrescu, Tabus, and Stoica 2001] and references therein). Specifically, it can be shown that an interior-point method (such as the ones in [Sturm 1999]), when applied to the SQP in (3.9.31), requires $\mathcal{O}(m^4)$ flops per iteration; furthermore, the number of iterations needed to achieve practical convergence of the method is typically quite small (and nearly independent of m), for instance between 10 and 20 iterations. The overall conclusion, therefore, is that (3.9.31), and hence *the original problem* (3.9.28)*, can be solved efficiently, in* $\mathcal{O}(m^4)$ *flops*. Once the solution to (3.9.31) has been computed, we can obtain the corresponding MA covariances either as $r = \hat{r} - C^{-1}\mu$ or as $r(k) = \mathrm{tr}(J_k Q)$ for $k = 0, \ldots, m$. Numerical results obtained with MA parameter estimation algorithm have been reported in [Dumitrescu, Tabus, and Stoica 2001]; see also [Stoica, McKelvey, and Mari 2000].

3.10 EXERCISES

Exercise 3.1: The Minimum Phase Property

As stated in the text, a polynomial $A(z)$ is said to be of minimum phase if all its zeroes are inside the unit circle. In this exercise, we motivate the name *minimum phase*. Specifically, we will show that, if $A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$ has real-valued coefficients and has all its zeroes inside the unit circle, and if $B(z)$ is any other polynomial in z^{-1} with real-valued coefficients that satisfies $|B(\omega)| = |A(\omega)|$ and $B(\omega = 0) = A(\omega = 0)$ (where $B(\omega) \triangleq B(z)|_{z=e^{i\omega}}$), then the phase *lag* of $B(\omega)$, given by $-\arg B(\omega)$, is greater than or equal to the phase lag of $A(\omega)$:

$$-\arg B(\omega) \geq -\arg A(\omega)$$

Since we can factor $A(z)$ as

$$A(z) = \prod_{k=1}^{n}(1 - \alpha_k z^{-1})$$

and $\arg A(\omega) = \sum_{k=1}^{n} \arg\left(1 - \alpha_k e^{-i\omega}\right)$, we begin by proving the minimum-phase property for first-order polynomials. Let

$$C(z) = 1 - \alpha z^{-1}, \qquad \alpha \triangleq re^{i\theta}, \quad r < 1$$

$$D(z) = z^{-1} - \alpha^* = C(z)\frac{z^{-1} - \alpha^*}{1 - \alpha z^{-1}} \triangleq C(z)E(z) \tag{3.10.1}$$

(a) Show that the zero of $D(z)$ is outside the unit circle, and that $|D(\omega)| = |C(\omega)|$.
(b) Show that

$$-\arg E(\omega) = \omega + 2\tan^{-1}\left[\frac{r\sin(\omega-\theta)}{1-r\cos(\omega-\theta)}\right]$$

Also, show that this function is increasing.
(c) If α is real, conclude that $-\arg D(\omega) \geq -\arg C(\omega)$ for $0 \leq \omega \leq \pi$, which justifies the name *minimum phase* for $C(z)$ in the first-order case.
(d) Generalize the first-order results proven in parts (a)–(c) to polynomials $A(z)$ and $B(z)$ of arbitrary order; in this case, the α_k either are real valued or occur in complex-conjugate pairs.

Exercise 3.2: Generating the ACS from ARMA Parameters
In this chapter, we developed equations expressing the ARMA coefficients $\{\sigma^2, a_i, b_j\}$ in terms of the ACS $\{r(k)\}_{k=-\infty}^{\infty}$. Find the inverse map; that is, given $\sigma^2, a_1, \ldots, a_n, b_1 \ldots, b_m$, find equations to determine $\{r(k)\}_{k=-\infty}^{\infty}$.

Exercise 3.3: Relationship between AR Modeling and Forward Linear Prediction
Suppose we have a zero-mean stationary process $\{y(t)\}$ (not necessarily AR) with ACS $\{r(k)\}_{k=-\infty}^{\infty}$. We wish to predict $y(t)$ by a linear combination of its n past values—that is, the predicted value is given by

$$\hat{y}_f(t) = \sum_{k=1}^{n} (-a_k) y(t-k)$$

We define the forward prediction error as

$$e_f(t) = y(t) - \hat{y}_f(t) = \sum_{k=0}^{n} a_k y(t-k)$$

with $a_0 = 1$. Show that the vector $\theta_f = [a_1 \ldots a_n]^T$ of prediction coefficients that minimizes the prediction-error variance $\sigma_f^2 \triangleq E\{|e_f(t)|^2\}$ is the solution to (3.4.2). Show also that $\sigma_f^2 = \sigma_n^2$ (i.e., that σ_n^2 in (3.4.2) is the prediction-error variance).

Furthermore, show that, if $\{y(t)\}$ is an AR(p) process with $p \leq n$, then the prediction error is white noise and that

$$k_j = 0 \qquad \text{for } j > p$$

where k_j is the jth reflection coefficient defined in (3.5.7). Show that, as a consequence, $a_{p+1}, \ldots, a_n = 0$. **Hint:** The calculations performed in Section 3.4.2 and in Complement 3.9.2 will be useful in solving this problem.

Exercise 3.4: Relationship between AR Modeling and Backward Linear Prediction
Consider the signal $\{y(t)\}$, as in Exercise 3.3. This time, we will consider backward prediction—that is, we will predict $y(t)$ from its n immediate future values:

$$\hat{y}_b(t) = \sum_{k=1}^{n} (-b_k) y(t+k)$$

This equation has corresponding backward prediction error $e_b(t) = y(t) - \hat{y}_b(t)$. Such backward prediction is useful in applications where noncausal processing is permitted—for example, when the data has been prerecorded and is stored in memory or on a tape and we want to make inferences on samples that precede the observed ones. Find an expression similar to (3.4.2) for the backward-prediction coefficient vector $\theta_b = [b_1 \ldots b_n]^T$. Find a relationship between the θ_b and the corresponding forward-prediction coefficient vector θ_f. Relate the forward and backward prediction-error variances.

Exercise 3.5: Prediction Filters and Smoothing Filters
The smoothing filter is a practically useful variation on the theme of linear prediction. A result of Exercises 3.3 and 3.4 should be that, for the forward and backward prediction filters

$$A(z) = 1 + \sum_{k=1}^{n} a_k z^{-k} \quad \text{and} \quad B(z) = 1 + \sum_{k=1}^{n} b_k z^{-k}$$

the prediction coefficients satisfy $a_k = b_k^*$ and the prediction-error variances are equal.

Now consider the *smoothing filter*

$$e_s(t) = \sum_{k=1}^{m} c_k y(t-k) + y(t) + \sum_{k=1}^{m} d_k y(t+k)$$

(a) Derive a system of linear equations, similar to the forward and backward linear-prediction equations, that relate the smoothing filter coefficients, the smoothing prediction-error variance $\sigma_s^2 = E\left\{|e_s(t)|^2\right\}$, and the ACS of $y(t)$.

(b) For $n = 2m$, provide an example of a zero-mean stationary random process for which the minimum smoothing prediction-error variance is *greater* than the minimum forward prediction-error variance. Also provide a second example where the minimum smoothing filter prediction-error variance is less than the corresponding minimum forward prediction-error variance.

(c) Assume $m = n$, but now constrain the smoothing prediction coefficients to be complex-conjugate symmetric: $c_k = d_k^*$ for $k = 1, \ldots, m$. In this case, the two prediction filters and the smoothing filter have the same number of degrees of freedom. Prove that the minimum smoothing prediction-error variance is less than or equal to the minimum (forward or backward) prediction-error variance. **Hint:** Show that the unconstrained minimum smoothing error variance solution (where we do not impose the constraint $c_k = d_k^*$) satisfies $c_k = d_k^*$ anyway.

Exercise 3.6: Relationship between Minimum Prediction Error and Spectral Flatness
Consider a random process $\{y(t)\}$, not necessarily an AR process, with ACS $\{r(k)\}$ and PSD $\phi_y(\omega)$. We find an AR(n) model for $y(t)$ by solving (3.4.6) for σ_n^2 and θ_n. These parameters generate an AR PSD model,

$$\phi_{AR}(\omega) = \frac{\sigma_n^2}{|A(\omega)|^2}$$

whose inverse Fourier transform we denote by $\{r_{AR}(k)\}_{k=-\infty}^{\infty}$. In this exercise, we explore the relationship between $\{r(k)\}$ and $\{r_{AR}(k)\}$ and that between $\phi_y(\omega)$ and $\phi_{AR}(\omega)$.

(a) Verify that the AR model has the property that

$$r_{AR}(k) = r(k), \qquad k = 0, \ldots, n.$$

(b) We have seen, from Exercise 3.3, that the AR model minimizes the nth-order forward prediction-error variance—that is, the variance of

$$e(t) = y(t) + a_1 y(t-1) + \ldots + a_n y(t-n)$$

For the special case that $\{y(t)\}$ is AR of order n or less, we also know that $\{e(t)\}$ is white noise, so $\phi_e(\omega)$ is flat. We will extend this last property by showing that, for general $\{y(t)\}$, $\phi_e(\omega)$ is maximally flat in the sense that the AR model maximizes the *spectral flatness* measure given by

$$f_e = \frac{\exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln \phi_e(\omega) d\omega\right]}{\frac{1}{2\pi}\int_{-\pi}^{\pi} \phi_e(\omega)\, d\omega} \tag{3.10.2}$$

where

$$\phi_e(\omega) = |A(\omega)|^2\, \phi_y(\omega) = \sigma_n^2 \frac{\phi_y(\omega)}{\phi_{AR}(\omega)}$$

Show that the measure f_e has the following "desirable" properties of a spectral flatness measure:
(i) f_e is unchanged if $\phi_e(\omega)$ is multiplied by a constant.
(ii) $0 \le f_e \le 1$.
(iii) $f_e = 1$ if and only if $\phi_e(\omega) =$ constant.

Hint: Use the fact that

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln |A(\omega)|^2\, d\omega = 0 \tag{3.10.3}$$

(This result can be proven by using the Cauchy integral formula). Show that (3.10.3) implies

$$f_e = f_y \frac{r_y(0)}{r_e(0)} \tag{3.10.4}$$

and thus that minimizing $r_e(0)$ maximizes f_e.

Exercise 3.7: Diagonalization of the Covariance Matrix
Show that R_{n+1} in equation (3.5.2) satisfies

$$L^* R_{n+1} L = D$$

where

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ & 1 & & \vdots & \vdots \\ & & \ddots & 0 & \\ & & & 1 & 0 \\ \theta_n & \theta_{n-1} & & \theta_1 & 1 \end{bmatrix} \quad \text{and} \quad D = \text{diag}\,[\sigma_n^2\ \sigma_{n-1}^2 \dots \sigma_0^2]$$

and where θ_k and σ_k^2 are defined in (3.4.6). Use this property to show that

$$|R_{n+1}| = \prod_{k=0}^{n} \sigma_k^2$$

Exercise 3.8: Stability of Yule–Walker AR Models
Assume that the matrix R_{n+1} in equation (3.4.6) is positive definite. (This can be achieved by using the sample covariances in (2.2.4) to build R_{n+1}, as explained in Section 2.2.) Then show that the AR model obtained from the Yule–Walker equations (3.4.6) is stable in the sense that the polynomial $A(z)$ has all its zeroes strictly inside the unit circle. (Most of the available proofs for this property are discussed in [Stoica and Nehorai 1987].)

Exercise 3.9: Three Equivalent Representations for AR Processes
In this chapter, we have considered three ways to parameterize an AR(n) process, but we have not explicitly shown when they are equivalent. Show that, for a nondegenerate AR(n) process (i.e., one for which R_{n+1} is positive definite), the following three parameterizations are equivalent:

(R) $r(0), \dots, r(n)$ such that R_{n+1} is positive definite.
(K) $r(0), k_1, \dots, k_n$ such that $r(0) > 0$ and $|k_i| < 1$ for $i = 1, \dots, n$.
(A) $\sigma_n^2, a_1, \dots, a_n$ such that $\sigma_n^2 > 0$ and all the zeroes of $A(z)$ are inside the unit circle.

Find the mapping from each parameterization to the others. (Some of these have already been derived in the text and in the previous exercises.)

Exercise 3.10: An Alternative Proof of the Stability Property of Reflection Coefficients
Prove that the $\hat{k}_p$ that minimizes (3.9.12) must be such that $|\hat{k}_p| \leq 1$, *without* using the expression (3.9.15) for $\hat{k}_p$. **Hint:** Write the criterion in (3.9.12) as

$$f(k_p) = \overline{E}\left(\left\| \begin{bmatrix} 1 & k_p \\ k_p^* & 1 \end{bmatrix} z(t) \right\|^2 \right)$$

where

$$\overline{E}(\cdot) = \frac{1}{2(N-p)} \sum_{t=p+1}^{N} (\cdot)$$

$$z(t) = \begin{bmatrix} \hat{e}_{f,p-1}(t) & \hat{e}_{b,p-1}(t-1) \end{bmatrix}^T$$

and show that if $|k_p| > 1$ then $f(k_p) > f(1/k_p^*)$.

Exercise 3.11: Recurrence Properties of the Reflection Coefficient Sequence for an MA Model
For an AR process of order n, the reflection coefficients satisfy $k_i = 0$ for $i > n$ (see Exercise 3.3) and the ACS satisfies the linear recurrence relationship $A(z)r(k) = 0$ for $k > 0$. Since an MA process of order m has the property that $r(i) = 0$ for $i > m$, we might wonder whether a recurrence relationship holds for the reflection coefficients corresponding to a MA process. We will investigate this "conjecture" for a simple case.

Consider an MA process of order 1 with parameter b_1. Show that $|R_n|$ satisfies the relationship

$$|R_n| = r(0)|R_{n-1}| - |r(1)|^2 |R_{n-2}|, \qquad n \geq 2$$

Show that $k_n = (-r(1))^n / |R_n|$ and that the reflection coefficient sequence satisfies the recurrence relationship

$$\frac{1}{k_n} = -\frac{r(0)}{r(1)} \frac{1}{k_{n-1}} - \frac{r^*(1)}{r(1)} \frac{1}{k_{n-2}} \tag{3.10.5}$$

with appropriate initial conditions (and state them). Show that the solution to (3.10.5) for $|b_1| < 1$ is

$$k_n = \frac{(1 - |b_1|^2)(-b_1)^n}{1 - |b_1|^{2n+2}} \tag{3.10.6}$$

This sequence decays exponentially to zero. When $b_1 = -1$, show that $k_n = 1/n$.

It has been shown that, for large n, $B(z)k_n \simeq 0$, where $\simeq 0$ means that the residue is small compared to the k_n terms [Georgiou 1987]. This result holds even for MA processes of order higher than 1. Unfortunately, the result is of little practical use as a means of estimating the b_k coefficients, because, for large n, the k_n values are (very) small.

Exercise 3.12: Asymptotic Variance of the ARMA Spectral Estimator
Consider the ARMA spectral estimator (3.2.2) with any consistent estimate of σ^2 and $\{a_i,\ b_j\}$. For simplicity, assume that the ARMA parameters are real; however, the result holds for complex ARMA processes as well. Show that the asymptotic (for large data sets) variance of this spectral estimator can be written in the form

$$E\left\{[\hat{\phi}(\omega) - \phi(\omega)]^2\right\} = C(\omega)\phi^2(\omega) \tag{3.10.7}$$

where $C(\omega) = \varphi^T(\omega)P\varphi(\omega)$. Here, P is the covariance matrix of the estimate of the parameter vector $[\sigma^2, a^T, b^T]^T$, and the vector $\varphi(\omega)$ has an expression that is to be found. Deduce that (3.10.7) has the same form as the asymptotic variance of the periodogram spectral estimator *but* with the essential difference that, in the ARMA-estimator case, $C(\omega)$ goes to zero as the number of data samples processed increases (and that $C(\omega)$ in (3.10.7) is a function of ω). **Hint:** Use a Taylor series expansion of $\hat{\phi}(\omega)$ as a function of the estimated parameters $\{\hat{\sigma}^2, \hat{a}_i, \hat{b}_j\}$. (See, for example, equation (B.1.1) in Appendix B.)

Exercise 3.13: Filtering Interpretation of Numerator Estimators in ARMA Estimation
An alternative method for estimating the MA part of an ARMA PSD is as follows: Assume we have estimated the AR coefficients (e.g., from equation (3.7.2) or (3.7.4)). We filter $y(t)$ by $\hat{A}(z)$ to form $f(t)$:

$$f(t) = y(t) + \sum_{i=1}^{n} \hat{a}_i y(t-i), \quad t = n+1, \ldots, N.$$

Then estimate the ARMA PSD as

$$\hat{\phi}(\omega) = \frac{\sum_{k=-m}^{m} \hat{r}_f(k)e^{-i\omega k}}{|\hat{A}(\omega)|^2}$$

where $\hat{r}_f(k)$ are the standard ACS estimates for $f(t)$. Show that this estimator is quite similar to (3.7.8) and (3.7.9) for large N.

Exercise 3.14: An Alternative Expression for ARMA Power Spectral Density
Consider an ARMA(n, m) process. Show that

$$\phi(z) = \sigma^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}$$

can be written as

$$\phi(z) = \frac{C(z)}{A(z)} + \frac{C^*(1/z^*)}{A^*(1/z^*)} \tag{3.10.8}$$

where

$$C(z) = \sum_{k=0}^{\max(m,n)} c_k z^{-k}$$

Show that the polynomial $C(z)$ satisfying (3.10.8) is unique, and find an expression for c_k in terms of $\{a_i\}$ and $\{r(k)\}$.

Equation (3.10.8) motivates an estimation procedure alternative to that in equations (3.7.8) and (3.7.9) for ARMA spectral estimation. In the alternative approach, we first estimate the AR coefficients $\{\hat{a}_i\}_{i=1}^n$ (using, for example, equation (3.7.2)). We then estimate the c_k coefficients, using the formula found in this exercise, and finally we insert the estimates $\hat{a}_k$ and $\hat{c}_k$ into the right-hand side of (3.10.8) to obtain a spectral estimate. Prove that this alternative estimator is equivalent to that in (3.7.8)–(3.7.9) under certain conditions, and find conditions on $\{\hat{a}_k\}$ so that they are equivalent. Also, compare (3.7.9) and (3.10.8) for ARMA(n, m) spectral estimation when $m < n$.

Exercise 3.15: Padé Approximation

A minimum-phase (or causally invertible) ARMA(n, m) model $B(z)/A(z)$ can be represented equivalently as an AR(∞) model $1/C(z)$. The approximation of a ratio of polynomials by a polynomial of higher order was considered by Padé in the late 1800s. One possible application of the Padé approximation is to obtain an ARMA spectral model by first estimating the coefficients of a high-order AR model, then solving for a (low-order) ARMA model from the estimated AR coefficients. In this exercise, we investigate the model relationships and some consequences of truncating the AR model polynomial coefficients.

Define

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$$
$$B(z) = 1 + b_1 z^{-1} + \cdots + b_m z^{-m}$$
$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots$$

(a) Show that

$$c_k = \begin{cases} 1, & k = 0 \\ a_k - \sum_{i=1}^m b_i c_{k-i}, & 1 \le k \le n \\ -\sum_{i=1}^m b_i c_{k-i}, & k > n \end{cases}$$

where we assume that any polynomial coefficient is equal to zero outside its defined range.

(b) Using the equations above, derive a procedure for computing the a_i and b_j parameters from a given set of $\{c_k\}_{k=0}^{m+n}$ parameters. Assume m and n are known.

(c) These equations give an exact representation using an infinite-order AR polynomial. In the Padé method, an *approximation* to $B(z)/A(z) = 1/C(z)$ is obtained by truncating (setting to zero) the c_k coefficients for $k > m + n$.
Suppose a stable minimum-phase ARMA(n, m) filter is approximated by an AR$(m + n)$ filter by using the Padé approximation. Give an example to show that the resulting AR approximation is not necessarily stable.

(d) Suppose a stable AR$(m + n)$ filter is approximated by a ratio $B_m(z)/A_n(z)$, as in part (b). Give an example to show that the resulting ARMA approximation is not necessarily stable.

Exercise 3.16: (Non)Uniqueness of Fully Parameterized ARMA Equations
The shaping filter (or transfer function) of the ARMA equation (3.8.1) is given by the *matrix fraction*

$$H(z) = A^{-1}(z)B(z), \qquad (ny \times ny) \tag{3.10.9}$$

where z is a dummy variable, and

$$A(z) = I + A_1 z^{-1} + \cdots + A_p z^{-p}$$
$$B(z) = I + B_1 z^{-1} + \cdots + B_p z^{-p}$$

(If the AR and MA orders, n and m, are different, then $p = \max(m, n)$.) Assume that $A(z)$ and $B(z)$ are "fully parameterized" in the sense that all elements of the matrix coefficients $\{A_i,\ B_j\}$ are unknown.

The matrix fraction description (MFD) (3.10.9) of the ARMA shaping filter is unique if and only if there exist *no* matrix polynomials $\tilde{A}(z)$ and $\tilde{B}(z)$ of degree p and *no* matrix polynomial $L(z) \neq I$ such that

$$\tilde{A}(z) = L(z)A(z) \qquad \tilde{B}(z) = L(z)B(z) \tag{3.10.10}$$

This can be verified by making use of (3.10.9). (See, for example, [KAILATH 1980].)

Show that the above uniqueness condition is satisfied for the fully parameterized MFD if and only if

$$\boxed{\text{rank}[A_p\ B_p] = ny} \tag{3.10.11}$$

Comment on the character of this condition: Is it restrictive or not?

COMPUTER EXERCISES

Tools for AR, MA, and ARMA Spectral Estimation:
The text website `www.prenhall.com/stoica` contains the following MATLAB functions for use in computing AR, MA, and ARMA spectral estimates and selecting the model order. For the first four functions, `y` is the input data vector, `n` is the desired AR order, and `m` is the desired MA order (if applicable). The outputs are `a`, the vector $[\hat{a}_1, \ldots, \hat{a}_n]^T$ of estimated AR parameters, `b`, the vector $[\hat{b}_1, \ldots, \hat{b}_m]^T$ of MA parameters (if applicable), and `sig2`, the noise variance estimate $\hat{\sigma}^2$. Variable definitions specific to particular functions are given here:

- `[a,sig2]=yulewalker(y,n)`
  The Yule–Walker AR method given by equation (3.4.2).
- `[a,sig2]=lsar(y,n)`
  The covariance least-squares AR method given by equation (3.4.12).

- `[a,gamma]=mywarma(y,n,m,M)`
  The modified Yule–Walker-based ARMA spectral estimate given by equation (3.7.9), where the AR coefficients are estimated from the overdetermined set of equations (3.7.4) with $W = I$. Here, `M` is the number of Yule–Walker equations used in (3.7.4), and `gamma` is the vector $[\hat{\gamma}_0, \ldots, \hat{\gamma}_m]^T$.
- `[a,b,sig2]=lsarma(y,n,m,K)`
  The two-stage least-squares ARMA method given in Section 3.7.2; `K` is the number of AR parameters to estimate in Step 1 of that algorithm.
- `order=armaorder(mo,sig2,N,nu)`
  Computes the AIC, AIC_c, GIC, and BIC model-order selections for general parameter-estimation problems. See Appendix C for details on the derivations of these methods. Here, `mo` is a vector of possible model orders, `sig2` is the vector of estimated residual variances corresponding to the model orders in `mo`, `N` is the length of the observed data vector, and `nu` is a parameter in the GIC method. The output 4-element vector `order` contains the model orders selected by using AIC, AIC_c, GIC, and BIC, respectively.

Exercise C3.17: Comparison of AR, ARMA, and Periodogram Methods for ARMA Signals In this exercise, we examine the properties of parametric methods for PSD estimation. We will use two ARMA signals, one broadband and one narrowband, to illustrate the performance of these parametric methods.

Broadband ARMA Process. Generate realizations of the broadband ARMA process

$$y(t) = \frac{B_1(z)}{A_1(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_1(z) = 1 - 1.3817z^{-1} + 1.5632z^{-2} - 0.8843z^{-3} + 0.4096z^{-4}$$
$$B_1(z) = 1 + 0.3544z^{-1} + 0.3508z^{-2} + 0.1736z^{-3} + 0.2401z^{-4}$$

Choose the number of samples as $N = 256$.

(a) Estimate the PSD of the realizations by using the four AR and ARMA estimators described above. Use AR(4), AR(8), ARMA(4,4), and ARMA(8,8); for the MYW algorithm, use both $M = n$ and $M = 2n$; for the LS AR(MA) algorithms, use $K = 2n$. Illustrate the performance by plotting ten overlaid estimates of the PSD. Also, plot the true PSD on the same diagram.

In addition, plot pole or pole–zero estimates for the various methods. (For the MYW method, the zeroes can be found by spectral factorization of the numerator; comment on the difficulties you encounter, if any.)

(b) Compare the two AR algorithms. How are they different in performance?

(c) Compare the two ARMA algorithms. How does M affect performance of the MYW algorithm? How do the accuracies of the respective pole and zero estimates compare?
(d) Use an ARMA(4,4) model for the LS ARMA algorithm, and estimate the PSD of the realizations for $K = 4$, 8, 12, and 16. How does K affect performance of the algorithm?
(e) Compare the lower-order estimates with the higher order estimates. In what way(s) does increasing the model order improve or degrade estimation performance?
(f) Compare the AR to the ARMA estimates. How does the AR(8) model perform with respect to the ARMA(4,4) model and the ARMA(8,8) model?
(g) Compare your results with those from the periodogram method on the same process (from Exercise C2.21 in Chapter 2). Comment on the difference between the methods with respect to variance, bias, and any other relevant properties of the estimators you notice.

Narrowband ARMA Process. Generate realizations of the narrowband ARMA process

$$y(t) = \frac{B_2(z)}{A_2(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_2(z) = 1 - 1.6408z^{-1} + 2.2044z^{-2} - 1.4808z^{-3} + 0.8145z^{-4}$$
$$B_2(z) = 1 + 1.5857z^{-1} + 0.9604z^{-2}$$

(a) Repeat the experiments and comparisons in the broadband example for the narrowband process; this time, use the following model orders: AR(4), AR(8), AR(12), AR(16), ARMA(4,2), ARMA(8,4), and ARMA(12,6).
(b) Study qualitatively how the algorithm performances differ for narrowband and broadband data. Comment separately on performance near the spectral peaks and near the spectral valleys.

Exercise C3.18: AR and ARMA Estimators for Line-Spectral Estimation
The ARMA methods can also be used to estimate line spectra. (Estimation of line spectra by other methods is the topic of Chapter 4.) In this application, AR(MA) techniques are often said to provide *superresolution* capabilities, because they are able to resolve sinusoids spaced too closely in frequency to be resolved by periodogram-based methods.

We again consider the four AR and ARMA estimators described previously.

(a) Generate realizations of the signal

$$y(t) = 10\sin(0.24\pi t + \varphi_1) + 5\sin(0.26\pi t + \varphi_2) + e(t), \qquad t = 1, \ldots, N$$

where $e(t)$ is (real) white Gaussian noise with variance σ^2 and where φ_1, φ_2 are independent random variables each uniformly distributed on $[0, 2\pi]$. From the results in Chapter 4, we

find the spectrum of $y(t)$ to be

$$\phi(\omega) = 50\pi\,[\delta(\omega - 0.24\pi) + \delta(\omega + 0.24\pi)] + 12.5\pi\,[\delta(\omega - 0.26\pi) + \delta(\omega + 0.26\pi)] + \sigma^2$$

(b) Compute the "true" AR polynomial (using the true ACS sequence; see equation (4.1.6)), using the Yule–Walker equations for both AR(4), AR(12), ARMA(4,4) and ARMA(12,12) models when $\sigma^2 = 1$. This experiment corresponds to estimates obtained as $N \to \infty$. Plot $1/|A(\omega)|^2$ for each case, and find the roots of $A(z)$. Which method(s) are able to resolve the two sinusoids?

(c) Consider now $N = 64$, and set $\sigma^2 = 0$; this corresponds to the case of finite data length but infinite SNR. Compute estimated AR polynomials, using the four spectral estimators and the AR and ARMA model orders described above; for the MYW technique, consider both $M = n$ and $M = 2n$, and, for the LS ARMA technique, use both $K = n$ and $K = 2n$. Plot $1/|A(\omega)|^2$, overlaid, for 50 different Monte Carlo simulations (using different values of φ_1 and φ_2 for each). Also, plot the zeroes of $A(z)$, overlaid, for these 50 simulations. Which method(s) are reliably able to resolve the sinusoids? Explain why. Note that as $\sigma^2 \to 0$, $y(t)$ corresponds to a (limiting) AR(4) process. How does the choice of M or K in the ARMA methods affect resolution or accuracy of the frequency estimates?

(d) Obtain spectral estimates ($\hat{\sigma}^2|\hat{B}(\omega)|^2/|\hat{A}(\omega)|^2$ for the ARMA estimators, and $\hat{\sigma}^2/|\hat{A}(\omega)|^2$ for the AR estimators) for the four methods when $N = 64$ and $\sigma^2 = 1$. Plot ten overlaid spectral estimates and overlaid polynomial zeroes of the $\hat{A}(z)$ estimates. Experiment with different AR and ARMA model orders to see whether the true frequencies are estimated more accurately; note also the appearance and severity of "spurious" sinusoids in the estimates for higher model orders. Which method(s) give reliable "superresolution" estimation of the sinusoids? How does the model order influence the resolution properties? Which method appears to have the best resolution?

You might want to experiment further by changing the SNR and the relative amplitudes of the sinusoids to gain a better understanding of the relative differences between the methods. Also, experiment with different model orders and parameters K and M to understand their impact on estimation accuracy.

(e) Compare the estimation results with periodogram-based estimates obtained from the same signals. Discuss differences in resolution, bias, and variance of the techniques.

Exercise C3.19: Model Order Selection for AR and ARMA Processes

In this exercise, we examine four methods for model order selection in AR and ARMA spectral estimation. We will experiment with both broadband and narrowband processes.

As discussed in Appendix C, several important model order selection rules have the general form (see (C.8.1)–(C.8.2))

$$-2\ln p_n(y, \hat{\theta}^n) + \eta(n, N)n \tag{3.10.12}$$

with different *penalty coefficients* $\eta(n, N)$ for the different methods:

$$\begin{aligned} \text{AIC}: &\quad \eta(n,N) = 2 \\ \text{AIC}_\text{c}: &\quad \eta(n,N) = 2\frac{N}{N-n-1} \\ \text{GIC}: &\quad \eta(n,N) = \nu \text{ (e.g., } \nu = 4) \\ \text{BIC}: &\quad \eta(n,N) = \ln N \end{aligned} \tag{3.10.13}$$

The term $\ln p_n(y, \hat{\theta}^n)$ is the log-likelihood of the observed data vector y, given the maximum-likelihood (ML) estimate of the parameter vector θ for a model of order n (where n is the total number of estimated real-valued parameters in the model); for the case of AR, MA, and ARMA models, a large-sample approximation for $-2\ln p_n(y, \hat{\theta}^n)$ that is commonly used for order selection (see, e.g., [Ljung 1987; Söderström and Stoica 1989]) is given by

$$-2\ln p_n(y, \hat{\theta}^n) \simeq N\hat{\sigma}_n^2 + \text{constant} \tag{3.10.14}$$

where $\hat{\sigma}_n^2$ is the sample estimate of σ^2 in (3.2.2) corresponding to the model of order n. The selected order is the value of n that minimizes (3.10.12). The order selection rules above, although derived for ML estimates of θ, can be used even with approximate ML estimates of θ, albeit with some loss of performance.

Broadband AR Process. Generate 100 realizations of the broadband AR process

$$y(t) = \frac{1}{A_1(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_1(z) = 1 - 1.3817z^{-1} + 1.5632z^{-2} - 0.8843z^{-3} + 0.4096z^{-4}$$

Choose the number of samples as $N = 128$. For each realization,

- **(a)** Estimate the model parameters, using the LS AR estimator and using AR model orders from 1 to 12.
- **(b)** Find the model orders that minimize the AIC, AIC_c, GIC (with $\nu = 4$), and BIC criteria. (See Appendix C.) Note that, for an AR model of order m, $n = m + 1$.
- **(c)** For each of the four order selection methods, plot a histogram of the selected orders for the 100 realizations. Comment on their relative performance.

Repeat this experiment, using $N = 256$ and $N = 1024$ samples. Discuss the relative performance of the order selection methods as N increases.

Narrowband AR Process. Repeat the previous experiment, using the narrowband AR process

$$y(t) = \frac{1}{A_2(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_2(z) = 1 - 1.6408z^{-1} + 2.2044z^{-2} - 1.4808z^{-3} + 0.8145z^{-4}$$

Compare the narrowband AR and broadband AR order selection results and discuss the relative order selection performance for these two AR processes.

Broadband ARMA Process. Repeat the broadband AR experiment by using the broadband ARMA process

$$y(t) = \frac{B_1(z)}{A_1(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_1(z) = 1 - 1.3817z^{-1} + 1.5632z^{-2} - 0.8843z^{-3} + 0.4096z^{-4}$$
$$B_1(z) = 1 + 0.3544z^{-1} + 0.3508z^{-2} + 0.1736z^{-3} + 0.2401z^{-4}$$

For the broadband ARMA process, use $N = 256$ and $N = 1024$ data samples. For each value of N, find ARMA(m, m) models (so $n = 2m + 1$ in equation (3.10.12)) for $m = 1, \ldots, 12$. Use the two-stage LS ARMA method with $K = 4m$ to estimate parameters.

Narrowband ARMA Process. Repeat the broadband ARMA experiment, but using the narrowband ARMA process

$$y(t) = \frac{B_2(z)}{A_2(z)}\, e(t)$$

with $\sigma^2 = 1$ and

$$A_2(z) = 1 - 1.6408z^{-1} + 2.2044z^{-2} - 1.4808z^{-3} + 0.8145z^{-4}$$
$$B_2(z) = 1 + 1.1100z^{-1} + 0.4706z^{-2}$$

Find ARMA$(2m, m)$ models for $m = 1, \ldots, 6$ (so $n = 3m + 1$ in equation (3.10.12)), using the two-stage LS ARMA method with $K = 8m$. Compare the narrowband ARMA and broadband ARMA order selection results and discuss the relative order selection performance for these two ARMA processes.

Exercise C3.20: AR and ARMA Estimators Applied to Measured Data
Consider the data sets in the files `sunspotdata.mat` and `lynxdata.mat`. These files can be obtained from the text website `www.prenhall.com/stoica`. Apply your favorite AR and ARMA estimator(s) (for the `lynx` data, use both the original data and the logarithmically transformed data, as in Exercise C2.23) to estimate the spectral content of these data. You will also need to select appropriate model orders m and n; see, for example, Exercise C3.19. As in Exercise C2.23, try to answer the following questions: Are there sinusoidal components (or periodic structure) in the data? If so, how many components and at what frequencies? Discuss the relative strengths and weaknesses of parametric and nonparametric estimators for understanding the spectral content of these data. In particular, discuss how a combination of the two techniques can be used to estimate the spectral and periodic structure of the data.