Appendix A

Linear Algebra and Matrix Analysis Tools

A.1 INTRODUCTION

In this appendix, we provide a review of the linear algebra terms and matrix properties used in the text. For the sake of brevity, we do not present proofs for all results stated in this appendix, nor do we discuss related results not needed in the chapters. For most of the results included, however, we do provide proofs and motivation. The reader interested in finding out more about the topic of this appendix can consult the books [Stewart 1973; Horn and Johnson 1985; Strang 1988; Horn and Johnson 1989; Golub and Van Loan 1989], to which we also refer for the proofs omitted here.

A.2 RANGE SPACE, NULL SPACE, AND MATRIX RANK

Let A be an $m \times n$ matrix whose elements are complex valued in general, $A \in \mathbf{C}^{m \times n}$, and let $(\cdot)^T$ and $(\cdot)^*$ denote the *transpose* and the *conjugate transpose* operator, respectively.

Definition D1: The ***range space*** of A, also called the ***column space***, is the subspace spanned by (all linear combinations of) the columns of A:

$$\mathcal{R}(A) = \{\alpha \in \mathbf{C}^{m\times 1} | \alpha = A\beta \quad \text{for} \quad \beta \in \mathbf{C}^{n\times 1}\} \tag{A.2.1}$$

The range space of A^T is usually called the ***row space*** of A, for obvious reasons.

Definition D2: The ***null space*** of A, also called the ***kernel***, is the following subspace:

$$\mathcal{N}(A) = \{\beta \in \mathbf{C}^{n\times 1} | A\beta = 0\} \tag{A.2.2}$$

The previous definitions are all that we need to introduce the matrix rank and its basic properties. We return to the range and null subspaces in Section A.4, where we discuss the singular-value decomposition. In particular, we derive some convenient bases and useful projectors associated with the previous matrix subspaces.

Definition D3: The following are equivalent definitions of the ***rank*** of A, denoted by

$$r \triangleq \text{rank}(A)$$

(i) r is equal to the maximum number of linearly independent columns of A. The latter number is by definition the dimension of the $\mathcal{R}(A)$; hence

$$r = \dim \mathcal{R}(A) \tag{A.2.3}$$

(ii) r is equal to the maximum number of linearly independent rows of A,

$$r = \dim \mathcal{R}(A^T) = \dim \mathcal{R}(A^*) \tag{A.2.4}$$

(iii) r is the dimension of the nonzero determinant of maximum size that can be built from the elements of A.

The equivalence between the preceding Definitions (i) and (ii) is an important and pleasing result (without which one should have had to consider the row rank and column rank of a matrix separately!).

Definition D4: A is said to be

- ***Rank deficient*** whenever $r < \min(m, n)$.
- ***Full column rank*** if $r = n \leq m$.
- ***Full row rank*** if $r = m \leq n$.
- ***Nonsingular*** whenever $r = m = n$.

Result R1: Premultiplication or postmultiplication of A by a nonsingular matrix does not change the rank of A.

Proof: This fact directly follows from the definition of rank(A), because the aforementioned multiplications do not change the number of linearly independent columns (or rows) of A. ■

Result R2: Let $A \in \mathbf{C}^{m\times n}$ and $B \in \mathbf{C}^{n\times p}$ be two conformable matrices of rank r_A and r_B, respectively. Then

$$\text{rank}(AB) \leq \min(r_A,\ r_B) \tag{A.2.5}$$

Proof: We can prove the previous assertion by using the definition of the rank once again. Indeed, premultiplication of B by A cannot increase the number of linearly independent columns of B, hence $\operatorname{rank}(AB) \leq r_B$. Similarly, postmultiplication of A by B cannot increase the number of linearly independent columns of A^T, which means that $\operatorname{rank}(AB) \leq r_A$. ■

Result R3: Let $A \in \mathbf{C}^{m \times m}$ be given by

$$A = \sum_{k=1}^{N} x_k \, y_k^*$$

where $x_k, \; y_k \in \mathbf{C}^{m \times 1}$. Then,

$$\operatorname{rank}(A) \leq \min(m, N)$$

Proof: A can be rewritten as

$$A = [x_1 \dots x_N] \begin{bmatrix} y_1^* \\ \vdots \\ y_N^* \end{bmatrix}$$

so the result follows from R2. ■

Result R4: Let $A \in \mathbf{C}^{m \times n}$ with $n \leq m$, let $B \in \mathbf{C}^{n \times p}$, and let

$$\operatorname{rank}(A) = n \tag{A.2.6}$$

Then

$$\operatorname{rank}(AB) = \operatorname{rank}(B) \tag{A.2.7}$$

Proof: Assumption (A.2.6) implies that A contains a nonsingular $n \times n$ submatrix, the postmultiplication of which by B gives a block of rank equal to $\operatorname{rank}(B)$ (*cf.* R1). Hence,

$$\operatorname{rank}(AB) \geq \operatorname{rank}(B)$$

However, by R2, $\operatorname{rank}(AB) \leq \operatorname{rank}(B)$; hence, (A.2.7) follows. ■

A.3 EIGENVALUE DECOMPOSITION

Definition D5: We say that the matrix $A \in \mathbf{C}^{m \times m}$ is ***Hermitian*** if $A^* = A$. In the real-valued case, such an A is said to be ***symmetric***.

Definition D6: A matrix $U \in \mathbf{C}^{m\times m}$ is said to be ***unitary*** (***orthogonal*** if U is real valued) whenever

$$U^*U = UU^* = I$$

If $U \in \mathbf{C}^{m\times n}$, with $m > n$, is such that $U^*U = I$, then we say that U is ***semiunitary***.

Next, we present a number of definitions and results pertaining to the matrix eigenvalue decomposition (EVD), first for general matrices and then for Hermitian ones.

A.3.1 General Matrices

Definition D7: A scalar $\lambda \in \mathbf{C}$ and a (nonzero) vector $x \in \mathbf{C}^{m\times 1}$ are an ***eigenvalue*** and its associated ***eigenvector*** of a matrix $A \in \mathbf{C}^{m\times m}$ if

$$Ax = \lambda x \tag{A.3.1}$$

In particular, an eigenvalue λ is a solution of the so-called ***characteristic equation*** of A, namely,

$$|A - \lambda I| = 0 \tag{A.3.2}$$

(where $|\cdot|$ denotes determinant) and x is a vector in $\mathcal{N}(A - \lambda I)$. The pair (λ, x) is called an ***eigenpair***.

Observe that, if $\{(\lambda_i, x_i)\}_{i=1}^{p}$ are p eigenpairs of A (with $p \le m$), then we can write the defining equations $Ax_i = \lambda x_i$ $(i = 1, \ldots, p)$ in the compact form

$$AX = X\Lambda \tag{A.3.3}$$

where

$$X = [x_1 \ldots x_p]$$

and

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \vdots & \\ 0 & & \lambda_p \end{bmatrix}$$

Result R5: Let (λ, x) be an eigenpair of $A \in \mathbf{C}^{m\times m}$. If $B = A + \alpha I$, with $\alpha \in \mathbf{C}$, then $(\lambda + \alpha, x)$ is an eigenpair of B.

Proof: The result follows from the fact that

$$Ax = \lambda x \implies (A + \alpha I)x = (\lambda + \alpha)x. \qquad \blacksquare$$

Result R6: The matrices A and $B \triangleq Q^{-1}AQ$, where Q is any nonsingular matrix, share the same eigenvalues. (B is said to be related to A by a ***similarity transformation***.)

Proof: Indeed, the equation

$$|B - \lambda I| = |Q^{-1}(A - \lambda I)Q| = |Q^{-1}||A - \lambda I||Q| = 0$$

is equivalent to $|A - \lambda I| = 0$. ■

In general, there is no simple relationship between the elements $\{A_{ij}\}$ of A and its eigenvalues $\{\lambda_k\}$. However, the *trace* of A, which is the sum of the diagonal elements of A, is related in a simple way to the eigenvalues, as described next.

Definition D8: The ***trace*** of a square matrix $A \in \mathbf{C}^{m\times m}$ is defined as

$$\operatorname{tr}(A) = \sum_{i=1}^{m} A_{ii} \tag{A.3.4}$$

Result R7: If $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of $A \in \mathbf{C}^{m\times m}$, then

$$\operatorname{tr}(A) = \sum_{i=1}^{m} \lambda_i \tag{A.3.5}$$

Proof: We can write

$$|\lambda I - A| = \prod_{i=1}^{n}(\lambda - \lambda_i) \tag{A.3.6}$$

The right-hand side of (A.3.6) is a polynomial in λ whose λ^{n-1} coefficient is $\sum_{i=1}^n \lambda_i$. From the definition of the determinant (see, e.g., [Strang 1988]), we find that the left-hand side of (A.3.6) is a polynomial whose λ^{n-1} coefficient is $\sum_{i=1}^n A_{ii} = \operatorname{tr}(A)$. This proves the result. ■

Interestingly, although the matrix product is not commutative, the trace is invariant to commuting the factors in a matrix product, as shown next.

Result R8: Let $A \in \mathbf{C}^{m\times n}$ and $B \in \mathbf{C}^{n\times m}$. Then

$$\operatorname{tr}(AB) = \operatorname{tr}(BA) \tag{A.3.7}$$

Proof: A straightforward calculation, based on the definition of tr(·) in (A.3.4), shows that

$$\begin{aligned}\operatorname{tr}(AB) &= \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ji} \\ &= \sum_{j=1}^{n}\sum_{i=1}^{m} B_{ji}A_{ij} = \sum_{j=1}^{n} [BA]_{jj} = \operatorname{tr}(BA)\end{aligned}$$ ■

We can also prove (A.3.7) by using Result R7. Along the way, we will obtain some other useful results. First, we note the following:

Result R9: Let $A, B \in \mathbf{C}^{m\times m}$ and let $\alpha \in \mathbf{C}$. Then

$$|AB| = |A|\,|B|$$
$$|\alpha A| = \alpha^m |A|$$

Proof: The identities follow directly from the definition of the determinant; see, for example, [Strang 1988]. ■

Next we prove the following results:

Result R10: Let $A \in \mathbf{C}^{m\times n}$ and $B \in \mathbf{C}^{n\times m}$. Then

$$|I - AB| = |I - BA|. \tag{A.3.8}$$

Proof: It is straightforward to verify that

$$\begin{bmatrix} I & A \\ 0 & I \end{bmatrix}\begin{bmatrix} I & -A \\ -B & I \end{bmatrix}\begin{bmatrix} I & 0 \\ B & I \end{bmatrix} = \begin{bmatrix} I - AB & 0 \\ 0 & I \end{bmatrix} \tag{A.3.9}$$

and

$$\begin{bmatrix} I & 0 \\ B & I \end{bmatrix}\begin{bmatrix} I & -A \\ -B & I \end{bmatrix}\begin{bmatrix} I & A \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I - BA \end{bmatrix} \tag{A.3.10}$$

Because the matrices in the left-hand sides of (A.3.9) and (A.3.10) have the same determinant, equal to $\begin{vmatrix} I & -A \\ -B & I \end{vmatrix}$, it follows that the right-hand sides must also have the same determinant, which concludes the proof. ■

Result R11: Let $A \in \mathbf{C}^{m\times n}$ and $B \in \mathbf{C}^{n\times m}$. The nonzero eigenvalues of AB and of BA are identical.

Proof: Let $\lambda \neq 0$ be an eigenvalue of AB. Then,

$$0 = |AB - \lambda I| = \lambda^m |AB/\lambda - I| = \lambda^m |BA/\lambda - I| = \lambda^{m-n} |BA - \lambda I|$$

where the third equality follows from R10. Hence, λ is also an eigenvalue of BA. ■

We can now obtain R8 as a simple corollary of R11, by using the property (A.3.5) of the trace operator.

A.3.2 Hermitian Matrices

An important property of the class of Hermitian matrices, which does not necessarily hold for general matrices, is the following:

Result R12:

(i) All eigenvalues of $A = A^* \in \mathbf{C}^{m \times m}$ are ***real valued***.
(ii) The m eigenvectors of $A = A^* \in \mathbf{C}^{m \times m}$ form an ***orthonormal set***. In other words, the matrix U, whose columns are the eigenvectors of A, is ***unitary***.

It follows from (i) and (ii) and from (A.3.3) that, for a Hermitian matrix, we can write

$$AU = U\Lambda$$

where $U^*U = UU^* = I$ and the diagonal elements of Λ are real numbers. Equivalently,

$$A = U\Lambda U^* \tag{A.3.11}$$

which is the so-called eigenvalue decomposition (EVD) of $A = A^*$. The EVD of a Hermitian matrix is a special case of the singular value decomposition of a general matrix, discussed in the next section.

The following is a useful result associated with Hermitian matrices:

Result R13: Let $A = A^* \in \mathbf{C}^{m \times m}$ and let $v \in \mathbf{C}^{m \times 1}$ $(v \neq 0)$. Also, let the eigenvalues of A be arranged in a nonincreasing order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$$

Then

$$\lambda_m \leq \frac{v^*Av}{v^*v} \leq \lambda_1 \tag{A.3.12}$$

The ratio in (A.3.12) is called the ***Rayleigh quotient***. Because this ratio is invariant under the multiplication of v by any complex number, we can rewrite (A.3.12) in the form:

$$\lambda_m \leq v^*Av \leq \lambda_1 \quad \textit{for any } v \in \mathbf{C}^{m\times 1} \textit{ with } v^*v = 1 \tag{A.3.13}$$

The equalities in (A.3.13) are evidently achieved when v is equal to the eigenvector of A associated with λ_m and λ_1, respectively.

Proof: Let the EVD of A be given by (A.3.11), and let

$$w = U^*v = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

We need to prove that

$$\lambda_m \leq w^*\Lambda w = \sum_{k=1}^{m} \lambda_k |w_k|^2 \leq \lambda_1$$

for any $w \in \mathbf{C}^{m\times 1}$ satisfying

$$w^*w = \sum_{k=1}^{m} |w_k|^2 = 1.$$

However, this is readily verified, as

$$\lambda_1 - \sum_{k=1}^{m} \lambda_k |w_k|^2 = \sum_{k=1}^{m} (\lambda_1 - \lambda_k)|w_k|^2 \geq 0$$

and

$$\sum_{k=1}^{m} \lambda_k |w_k|^2 - \lambda_m = \sum_{k=1}^{m} (\lambda_k - \lambda_m)|w_k|^2 \geq 0$$

and the proof is concluded. ■

The following result is an extension of R13.

Result R14: Let $V \in \mathbf{C}^{m\times n}$, with $m > n$, be a semiunitary matrix (i.e., $V^*V = I$), and let $A = A^* \in \mathbf{C}^{m\times m}$ have its eigenvalues ordered as in R13. Then

$$\sum_{k=m-n+1}^{m} \lambda_k \leq \mathrm{tr}(V^*AV) \leq \sum_{k=1}^{n} \lambda_k \tag{A.3.14}$$

where the equalities are achieved, for instance, when the columns of V are the eigenvectors of A corresponding to $(\lambda_{m-n+1}, \ldots, \lambda_m)$ and, respectively, to $(\lambda_1, \ldots, \lambda_n)$. The ratio

$$\frac{\operatorname{tr}(V^*AV)}{\operatorname{tr}(V^*V)} = \frac{\operatorname{tr}(V^*AV)}{n}$$

is sometimes called the ***extended Rayleigh quotient***.

Proof: Let

$$A = U\Lambda U^*$$

(*cf.* (A.3.11)), and let

$$S = U^*V \triangleq \begin{bmatrix} s_1^* \\ \vdots \\ s_m^* \end{bmatrix} \qquad (m \times n)$$

(hence, s_k^* is the kth row of S). By making use of the preceding notation, we can write

$$\operatorname{tr}(V^*AV) = \operatorname{tr}(V^*U\Lambda U^*V) = \operatorname{tr}(S^*\Lambda S) = \operatorname{tr}(\Lambda SS^*) = \sum_{k=1}^{m} \lambda_k c_k \tag{A.3.15}$$

where

$$c_k \triangleq s_k^* s_k, \qquad k = 1, \ldots m \tag{A.3.16}$$

Clearly,

$$c_k \geq 0, \qquad k = 1, \ldots, m \tag{A.3.17}$$

and

$$\sum_{k=1}^{m} c_k = \operatorname{tr}(SS^*) = \operatorname{tr}(S^*S) = \operatorname{tr}(V^*UU^*V) = \operatorname{tr}(V^*V) = \operatorname{tr}(I) = n \tag{A.3.18}$$

Furthermore,

$$c_k \leq 1, \qquad k = 1, \ldots, m. \tag{A.3.19}$$

To see this, let $G \in \mathbf{C}^{m\times(m-n)}$ be such that the matrix $[S \; G]$ is unitary; and let g_k^* denote the kth row of G. Then, by construction,

$$[s_k^* \; g_k^*] \begin{bmatrix} s_k \\ g_k \end{bmatrix} = c_k + g_k^* g_k = 1 \implies c_k = 1 - g_k^* g_k \leq 1$$

which is (A.3.19).

Finally, by combining (A.3.15) with (A.3.17)–(A.3.19), we can readily verify that $\operatorname{tr}(V^*AV)$ satisfies (A.3.14), where the equalities are achieved for

$$c_1 = \cdots = c_{m-n} = 0; \quad c_{m-n+1} = \cdots = c_m = 1$$

and, respectively,

$$c_1 = \cdots = c_n = 1; \quad c_{n+1} = \cdots = c_m = 0$$

These conditions on $\{c_k\}$ are satisfied if, for example, S is equal to $[0\ I]^T$ and $[I\ 0]^T$, respectively. With this observation, the proof is concluded. ■

Result R13 is clearly a special case of Result R14. The only reason for considering R13 separately is that the simpler result R13 is used more often in the text than R14.

A.4 SINGULAR VALUE DECOMPOSITION AND PROJECTION OPERATORS

For any matrix $A \in \mathbf{C}^{m\times n}$, there exist unitary matrices $U \in \mathbf{C}^{m\times m}$ and $V \in \mathbf{C}^{n\times n}$ and a diagonal matrix $\Sigma \in \mathbf{R}^{m\times n}$ with nonnegative diagonal elements, such that

$$A = U\Sigma V^* \tag{A.4.1}$$

By appropriate permutation, the diagonal elements of Σ can be arranged in a nonincreasing order:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)}$$

The factorization (A.4.1) is called the ***singular value decomposition*** (SVD) of A, and its existence is a significant result both from a theoretical and from a practical standpoint. We reiterate that the matrices U, Σ, and V in (A.4.1) satisfy the equations

$$\begin{aligned} U^*U &= UU^* = I \quad (m \times m) \\ V^*V &= VV^* = I \quad (n \times n) \\ \Sigma_{ij} &= \begin{cases} \sigma_i \geq 0 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \end{aligned}$$

The following terminology is most commonly associated with the SVD:

- The ***left singular vectors*** of A are the columns of U. These singular vectors are also the eigenvectors of the matrix AA^*.
- The ***right singular vectors*** of A are the columns of V. These vectors are also the eigenvectors of the matrix A^*A.
- The ***singular values*** of A are the diagonal elements $\{\sigma_i\}$ of Σ. Note that $\{\sigma_i\}$ are the square roots of the largest $\min(m, n)$ eigenvalues of AA^* or A^*A.

- The ***singular triple*** of A is the following triple: (singular value, left singular vector, and right singular vector; σ_k, u_k, v_k), where u_k (v_k) is the kth column of $U(V)$.

If

$$\operatorname{rank}(A) = r \le \min(m, n)$$

then one can show that

$$\begin{cases} \sigma_k > 0, & k = 1, \ldots, r \\ \sigma_k = 0, & k = r + 1, \ldots, \min(m, n) \end{cases}$$

Hence, for a matrix of rank r, the SVD can be written as

$$A = [\,\underbrace{U_1}_{r} \quad \underbrace{U_2}_{m-r}\,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \begin{matrix} \}\, r \\ \}\, n-r \end{matrix} = U_1 \Sigma_1 V_1^* \tag{A.4.2}$$

where $\Sigma_1 \in \mathbf{R}^{r \times r}$ is nonsingular. The factorization of A in (A.4.2) has a number of important consequences.

Result R15: Consider the SVD of $A \in \mathbf{C}^{m \times n}$ in (A.4.2), where $r \le \min(m, n)$. Then

(i) U_1 is an orthonormal basis of $\mathcal{R}(A)$;
(ii) U_2 is an orthonormal basis of $\mathcal{N}(A^*)$;
(iii) V_1 is an orthonormal basis of $\mathcal{R}(A^*)$;
(iv) V_2 is an orthonormal basis of $\mathcal{N}(A)$.

Proof: We see that (iii) and (iv) follow from the properties (i) and (ii) as applied to A^*. To prove (i) and (ii), we need to show that

$$\mathcal{R}(A) = \mathcal{R}(U_1) \tag{A.4.3}$$

and, respectively,

$$\mathcal{N}(A^*) = \mathcal{R}(U_2) \tag{A.4.4}$$

To show (A.4.3), note that

$$\alpha \in \mathcal{R}(A) \Rightarrow \text{there exists } \beta \text{ such that } \alpha = A\beta \Rightarrow$$
$$\Rightarrow \alpha = U_1(\Sigma_1 V_1^* \beta) = U_1 \gamma \Rightarrow \alpha \in \mathcal{R}(U_1)$$

so $\mathcal{R}(A) \subset \mathcal{R}(U_1)$. Also,

$$\alpha \in \mathcal{R}(U_1) \Rightarrow \text{there exists } \beta \text{ such that } \alpha = U_1 \beta$$

From (A.4.2), $U_1 = AV_1\Sigma_1^{-1}$; it follows that

$$\alpha = A(V_1\Sigma_1^{-1}\beta) = A\rho \Rightarrow \alpha \in \mathcal{R}(A)$$

which shows $\mathcal{R}(U_1) \subset \mathcal{R}(A)$. Combining $\mathcal{R}(U_1) \subset \mathcal{R}(A)$ with $\mathcal{R}(A) \subset \mathcal{R}(U_1)$ gives (A.4.3). Similarly,

$$\alpha \in \mathcal{N}(A^*) \Rightarrow A^*\alpha = 0 \Rightarrow V_1\Sigma_1 U_1^*\alpha = 0 \Rightarrow \Sigma_1^{-1}V_1^*V_1\Sigma_1 U_1^*\alpha = 0 \Rightarrow U_1^*\alpha = 0$$

Now, any vector α can be written as

$$\alpha = [U_1 \;\; U_2]\begin{bmatrix} \gamma \\ \beta \end{bmatrix}$$

because $[U_1 \;\; U_2]$ is nonsingular. However, $0 = U_1^*\alpha = U_1^*U_1\gamma + U_1^*U_2\beta = \gamma$, so $\gamma = 0$, and thus $\alpha = U_2\beta$. Thus, $\mathcal{N}(A^*) \subset \mathcal{R}(U_2)$. Finally,

$$\alpha \in \mathcal{R}(U_2) \Rightarrow \text{there exists } \beta \text{ such that } \alpha = U_2\beta$$

Then

$$A^*\alpha = V_1\Sigma_1 U_1^* U_2\beta = 0 \Rightarrow \alpha \in \mathcal{N}(A^*)$$

which leads to (A.4.4). ■

This result, readily derived by using the SVD, has a number of interesting corollaries that complement the discussion on range and null subspaces in Section A.2.

Result R16: For any $A \in \mathbf{C}^{m\times n}$, the subspaces $\mathcal{R}(A)$ and $\mathcal{N}(A^*)$ are orthogonal to each other, and they together span $\mathbf{C}^m$. Consequently, we say that $\mathcal{N}(A^*)$ is the ***orthogonal complement*** of $\mathcal{R}(A)$ in $\mathbf{C}^m$, and vice versa. In particular, we have

$$\dim \mathcal{N}(A^*) = m - r \qquad \text{(A.4.5)}$$

$$\dim \mathcal{N}(A) = n - r \qquad \text{(A.4.6)}$$

(Recall that $\dim \mathcal{R}(A) = \dim \mathcal{R}(A^*) = r$.)

Proof: This result is a direct corollary of R15. ■

The SVD of a matrix also provides a convenient representation for the projectors onto the range and null spaces of A and A^*.

Definition D9: Let $y \in \mathbf{C}^{m\times 1}$ be an arbitrary vector. By definition, the ***orthogonal projector*** onto $\mathcal{R}(A)$ is the matrix Π, which is such that (i) $\mathcal{R}(\Pi) = \mathcal{R}(A)$ and (ii) the Euclidean distance between y and $\Pi y \in \mathcal{R}(A)$ is minimum:

$$\|y - \Pi y\|^2 = \min \quad \text{over } \mathcal{R}(A)$$

Hereafter, $\|x\| = \sqrt{x^* x}$ denotes the ***Euclidean vector norm***.

Result R17: Let $A \in \mathbf{C}^{m\times n}$. The orthogonal projector onto $\mathcal{R}(A)$ is given by

$$\Pi = U_1 U_1^* \tag{A.4.7}$$

whereas the orthogonal projector onto $\mathcal{N}(A^*)$ is

$$\Pi^{\perp} = I - U_1 U_1^* = U_2 U_2^* \tag{A.4.8}$$

Proof: Let $y \in \mathbf{C}^{m\times 1}$ be an arbitrary vector. As $\mathcal{R}(A) = \mathcal{R}(U_1)$, according to R15, we can find the vector in $\mathcal{R}(A)$ that is of minimal distance from y by solving the problem

$$\min_{\beta} \|y - U_1\beta\|^2 \tag{A.4.9}$$

Because

$$\begin{aligned}\|y - U_1\beta\|^2 &= (\beta^* - y^* U_1)(\beta - U_1^* y) + y^*(I - U_1 U_1^*)y \\ &= \|\beta - U_1^* y\|^2 + \|U_2^* y\|^2\end{aligned}$$

it readily follows that the solution to the minimization problem (A.4.9) is given by $\beta = U_1^* y$. Hence, the vector $U_1 U_1^* y$ is the orthogonal projection of y onto $\mathcal{R}(A)$, and the minimum distance from y to $\mathcal{R}(A)$ is $\|U_2^* y\|$. This proves (A.4.7). Then (A.4.8) follows immediately from (A.4.7) and the fact that $\mathcal{N}(A^*) = \mathcal{R}(U_2)$. ■

Note, for instance, that, for the projection of y onto $\mathcal{R}(A)$, the error vector is $y - U_1 U_1^* y = U_2 U_2^* y$, which is in $\mathcal{R}(U_2)$ and therefore is orthogonal to $\mathcal{R}(A)$ by R15. For this reason, Π is given the name "orthogonal projector" in D9 and R17.

As an aside, we remark that the orthogonal projectors in (A.4.7) and (A.4.8) are *idempotent matrices*; see the next definition.

Definition D10: The matrix $A \in \mathbf{C}^{m\times m}$ is ***idempotent*** if

$$A^2 = A \tag{A.4.10}$$

Furthermore, observe, by making use of R11, that the idempotent matrix in (A.4.7), for example, has r eigenvalues equal to 1 and $(m - r)$ eigenvalues equal to 0. This is a general property of idempotent matrices: their eigenvalues are either 0 or 1.

Finally, we present a result that, even alone, would be enough to make the SVD an essential matrix-analysis tool.

Result R18: Let $A \in \mathbf{C}^{m\times n}$, with elements A_{ij}. Let the SVD of A (with the singular values arranged in a nonincreasing order) be given by

$$A = [\,\underbrace{U_1}_{p} \quad \underbrace{U_2}_{m-p}\,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \begin{matrix} \}\, p \\ \}\, n-p \end{matrix} \tag{A.4.11}$$

where $p \leq \min(m, n)$ is an integer. Let

$$\|A\|^2 = \operatorname{tr}(A^*A) = \sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2 = \sum_{k=1}^{\min(m,n)} \sigma_k^2 \tag{A.4.12}$$

denote the square of the so-called ***Frobenius norm***. Then the ***best rank-p approximant*** of A in the Frobenius-norm metric, that is, the solution to

$$\min_{B} \ \|A - B\|^2 \quad \text{subject to } \operatorname{rank}(B) = p \ , \tag{A.4.13}$$

is given by

$$B_0 = U_1 \Sigma_1 V_1^* \tag{A.4.14}$$

Furthermore, B_0 is the unique solution to the approximation problem (A.4.13) if and only if $\sigma_p > \sigma_{p+1}$.

Proof: It follows from R4 and (A.4.2) that we can parameterize B in (A.4.13) as

$$B = CD^* \tag{A.4.15}$$

where $C \in \mathbf{C}^{m\times p}$ and $D \in \mathbf{C}^{n\times p}$ are full-column-rank matrices. The previous parameterization of B is of course nonunique, but, as we will see, this fact does not introduce any problem. By making use of (A.4.15), we can rewrite the problem (A.4.13) in the following form:

$$\min_{C,D} \ \|A - CD^*\|^2 \quad \operatorname{rank}(C) = \operatorname{rank}(D) = p \tag{A.4.16}$$

The reparameterized problem is essentially constraint free. Indeed, the full-column-rank condition that must be satisfied by C and D can be easily handled.

First, we minimize (A.4.16) with respect to D, for a given C. To that end, observe that

$$\begin{aligned} \|A - CD^*\|^2 = \operatorname{tr}\{&[D - A^*C(C^*C)^{-1}](C^*C)[D^* - (C^*C)^{-1}C^*A] \\ &+ A^*[I - C(C^*C)^{-1}C^*]A\} \end{aligned} \tag{A.4.17}$$

By result (iii) in Definition D11 in the next section, the matrix $[D - A^*C(C^*C)^{-1}] \cdot (C^*C)[D^* - (C^*C)^{-1}C^*A]$ is positive semidefinite for any D. This observation implies that (A.4.17) is minimized with respect to D for

$$D_0 = A^*C(C^*C)^{-1} \tag{A.4.18}$$

and that the corresponding minimum value of (A.4.17) is given by

$$\operatorname{tr}\{A^*[I - C(C^*C)^{-1}C^*]A\} \tag{A.4.19}$$

Next, we minimize (A.4.19) with respect to C. Let $S \in \mathbf{C}^{m \times p}$ denote an orthogonal basis of $\mathcal{R}(C)$—that is, $S^*S = I$ and

$$S = C\Gamma$$

for some nonsingular $p \times p$ matrix Γ. It is then straightforward to verify that

$$I - C(C^*C)^{-1}C^* = I - SS^* \tag{A.4.20}$$

By combining (A.4.19) and (A.4.20), we can restate the problem of minimizing (A.4.19) with respect to C as

$$\max_{S;\, S^*S=I} \operatorname{tr}[S^*(AA^*)S] \tag{A.4.21}$$

The solution to (A.4.21) follows from R14; the maximizing S is given by

$$S_0 = U_1$$

which yields

$$C_0 = U_1\Gamma^{-1} \tag{A.4.22}$$

It follows that

$$\begin{aligned} B_0 &= C_0 D_0^* = C_0(C_0^*C_0)^{-1}C_0^*A = S_0S_0^*A \\ &= U_1U_1^*(U_1\Sigma_1V_1^* + U_2\Sigma_2V_2^*) \\ &= U_1\Sigma_1V_1^*. \end{aligned}$$

Furthermore, we observe that the minimum value of the Frobenius distance in (A.4.13) is given by

$$\|A - B_0\|^2 = \|U_2\Sigma_2V_2^*\|^2 = \sum_{k=p+1}^{\min(m,n)} \sigma_k^2$$

If $\sigma_p > \sigma_{p+1}$, then the best rank-p approximant B_0 is unique; otherwise, it is not unique. Indeed, whenever $\sigma_p = \sigma_{p+1}$, we can obtain B_0 by using either the singular vectors associated with σ_p or those corresponding to σ_{p+1}; each alternative choice generally leads to a different solution.

■

A.5 POSITIVE (SEMI)DEFINITE MATRICES

Let $A = A^* \in \mathbf{C}^{m\times m}$ be a Hermitian matrix, and let $\{\lambda_k\}_{k=1}^m$ denote its eigenvalues.

Definition D11: We say that A is ***positive semidefinite*** (psd) or ***positive definite*** (pd) if any of the following equivalent conditions holds true:

(i) $\lambda_k \geq 0$ ($\lambda_k > 0$ for pd) for $k = 1, \ldots, m$.

(ii) $\alpha^* A\alpha \geq 0$ ($\alpha^* A\alpha > 0$ for pd) for any nonzero vector $\alpha \in \mathbf{C}^{m\times 1}$

(iii) There exists a matrix C such that

$$A = CC^* \tag{A.5.1}$$

(with $\operatorname{rank}(C) = m$ for pd)

(iv) $|A(i_1, \ldots, i_k)| \geq 0$ (> 0 for pd) for all $k = 1, \ldots, m$ and all indices $i_1, \ldots, i_k \in [1, m]$, where $A(i_1, \ldots, i_k)$ is the submatrix formed from A by eliminating the $i_1, \ldots, i_k$ rows and columns of A. ($A(i_1, \ldots, i_k)$ is called a ***principal submatrix*** of A). The condition for A to be positive definite can be simplified to requiring that $|A(k+1, \ldots, m)| > 0$ (for $k = 1, \ldots, m-1$) and $|A| > 0$. ($A(k+1, \ldots, m)$ is called a ***leading submatrix*** of A).

The notation $A > 0$ ($A \geq 0$) is commonly used to denote that A is pd (psd).

Of the previous defining conditions, (iv) is apparently the most involved. The necessity of (iv) can be proven as follows: Let α be a vector in $\mathbf{C}^m$ with zeroes at the positions $\{i_1, \ldots, i_k\}$ and arbitrary elements elsewhere. Then, by using (ii), we readily see that $A \geq 0$ (> 0) implies $A(i_1, \ldots, i_k) \geq 0$ (> 0), which, in turn, implies (iv) by making use of (i) and the fact that the determinant of a matrix equals the product of its eigenvalues. The sufficiency of (iv) is shown in [STRANG 1988].

The equivalence of the remaining conditions, (i), (ii), and (iii), is easily proven by making use of the EVD of A: $A = U\Lambda U^*$. To show that (i) $\Leftrightarrow$ (ii), assume first that (i) holds and let $\beta = U^*\alpha$. Then

$$\alpha^* A\alpha = \beta^*\Lambda\beta = \sum_{k=1}^{m} \lambda_k |\beta_k|^2 \geq 0 \tag{A.5.2}$$

and hence, (ii) holds as well. Conversely, because U is invertible, it follows from (A.5.2) that (ii) can hold, only if (i) holds; indeed, if (i) does not hold, one can choose β to make (A.5.2) negative; thus there exists an $\alpha = U\beta$ such that $\alpha^* A\alpha < 0$, which contradicts the assumption that (ii) holds. Consequently, (i) and (ii) are equivalent. To show that (iii) $\Rightarrow$ (ii), note that

$$\alpha^* A\alpha = \alpha^* CC^*\alpha = \|C^*\alpha\|^2 \geq 0$$

and thus (ii) holds as well. Because (iii) $\Rightarrow$ (ii) and (ii) $\Rightarrow$ (i), we have (iii) $\Rightarrow$ (i). To show that (i) $\Rightarrow$ (iii), we assume (i) and write

$$A = U\Lambda U^* = (U\Lambda^{1/2}\Lambda^{1/2}U^*) = (U\Lambda^{1/2}U^*)(U\Lambda^{1/2}U^*) \triangleq CC^* \tag{A.5.3}$$

and thus (iii) is also satisfied. In (A.5.3), $\Lambda^{1/2}$ is a diagonal matrix whose diagonal elements are equal to $\{\lambda_k^{1/2}\}$. In other words, $\Lambda^{1/2}$ is the "square root" of Λ.

In a general context, the square root of a positive semidefinite matrix is defined as follows:

Definition D12: Let $A = A^*$ be a positive semidefinite matrix. Then any matrix C that satisfies

$$A = CC^* \tag{A.5.4}$$

is called a ***square root*** of A. Sometimes such a C is denoted by $A^{1/2}$.

If C is a square root of A, then so is CB for any unitary matrix B; hence, a given positive semidefinite matrix has an infinite number of square roots. Two often-used particular choices for square roots are

(i) *Hermitian square root*: $C = C^*$. In this case, we can write (A.5.4) as $A = C^2$. Note that we have already obtained such a square root of A in (A.5.3):

$$C = U\Lambda^{1/2}U^* \tag{A.5.5}$$

If C is also constrained to be positive semidefinite ($C \geq 0$) then the Hermitian square root is unique.

(ii) *Cholesky factor*. If C is lower triangular with nonnegative diagonal elements, then C is called the *Cholesky factor* of A. In computational exercises, the triangular form of the square-root matrix is often preferred to other forms. If A is positive definite, the Cholesky factor is unique.

We also note that equation (A.5.4) implies that A and C *have the same rank and the same range space*. This follows easily, for example, from inserting the SVD of C into (A.5.4).

Next, we prove three specialized results on positive semidefinite matrices required in Section 2.5 and in Appendix B.

Result R19: Let $A \in \mathbf{C}^{m\times m}$ and $B \in \mathbf{C}^{m\times m}$ be positive semidefinite matrices. Then the matrix $A \odot B$ is also positive semidefinite, where $\odot$ denotes the ***Hadamard matrix product*** (also called ***elementwise multiplication***: $[A \odot B]_{ij} = A_{ij}B_{ij}$).

Proof: Because B is positive semidefinite, it can be written as $B = CC^*$ for some matrix $C \in \mathbf{C}^{m\times m}$. Let c_k^* denote the kth row of C. Then,

$$[A \odot B]_{ij} = A_{ij}B_{ij} = A_{ij}\ c_i^* c_j$$

and, hence, for any $\alpha \in \mathbf{C}^{m\times 1}$,

$$\alpha^*(A \odot B)\alpha = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^* A_{ij} c_i^* c_j \alpha_j \tag{A.5.6}$$

By letting $\{c_{jk}\}_{k=1}^{m}$ denote the elements of the vector c_j, we can rewrite (A.5.6) as

$$\alpha^*(A \odot B)\alpha = \sum_{k=1}^{m} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i^* c_{ik}^* A_{ij} \alpha_j c_{jk} = \sum_{k=1}^{m} \beta_k^* A \beta_k \tag{A.5.7}$$

where

$$\beta_k \triangleq [\alpha_1 c_{1k} \cdots \alpha_m c_{mk}]^T$$

A is positive semidefinite by assumption, so $\beta_k^* A \beta_k \geq 0$ for each k, and it follows from (A.5.7) that $A \odot B$ must be positive semidefinite as well. ■

Result R20: Let $A \in \mathbf{C}^{m\times m}$ and $B \in \mathbf{C}^{m\times m}$ be Hermitian matrices. Assume that B is nonsingular and that the partitioned matrix

$$\begin{bmatrix} A & I \\ I & B \end{bmatrix}$$

is positive semidefinite. Then the matrix $(A - B^{-1})$ is also positive semidefinite:

$$A \geq B^{-1}$$

Proof: By Definition D11, part (ii),

$$\begin{bmatrix} \alpha_1^* & \alpha_2^* \end{bmatrix} \begin{bmatrix} A & I \\ I & B \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \geq 0 \tag{A.5.8}$$

for any vectors $\alpha_1, \alpha_2 \in \mathbf{C}^{m\times 1}$. Let

$$\alpha_2 = -B^{-1}\alpha_1$$

Then (A.5.8) becomes

$$\alpha_1^*(A - B^{-1})\alpha_1 \geq 0$$

This inequality must hold for any $\alpha_1 \in \mathbf{C}^{m\times 1}$, and so the proof is concluded. ■

Result R21: Let $C \in \mathbf{C}^{m\times m}$ be a (Hermitian) positive definite matrix depending on a real-valued parameter α. Assume that C is a differentiable function of α. Then

$$\frac{\partial}{\partial\alpha}[\ln |C|] = \operatorname{tr}\left[C^{-1}\frac{\partial C}{\partial\alpha}\right]$$

Proof: Let $\{\lambda_i\} \in \mathbf{R}$ $(i = 1, \dots, m)$ denote the eigenvalues of C. Then

$$\begin{aligned}
\frac{\partial}{\partial \alpha} [\ln |C|] = \frac{\partial}{\partial \alpha} \left[\ln \prod_{k=1}^{m} \lambda_k \right] &= \sum_{k=1}^{m} \frac{\partial}{\partial \alpha} (\ln \lambda_k) \\
&= \sum_{k=1}^{m} \frac{1}{\lambda_k} \frac{\partial \lambda_k}{\partial \alpha} \\
&= \operatorname{tr} \left[\Lambda^{-1} \frac{\partial \Lambda}{\partial \alpha} \right]
\end{aligned}$$

where $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_m)$. Let Q be a unitary matrix such that $Q^* \Lambda Q = C$ (which is the EVD of C). Since Q is unitary, $Q^* Q = I$, we obtain

$$\frac{\partial Q^*}{\partial \alpha} Q + Q^* \frac{\partial Q}{\partial \alpha} = 0$$

Thus, we get

$$\begin{aligned}
\operatorname{tr} \left[\Lambda^{-1} \frac{\partial \Lambda}{\partial \alpha} \right] &= \operatorname{tr} \left[\left(Q^* \Lambda^{-1} Q \right) \left(Q^* \frac{\partial \Lambda}{\partial \alpha} Q \right) \right] \\
&= \operatorname{tr} \left[C^{-1} \left(\frac{\partial}{\partial \alpha} \left(Q^* \Lambda Q \right) - \frac{\partial Q^*}{\partial \alpha} \Lambda Q - Q^* \Lambda \frac{\partial Q}{\partial \alpha} \right) \right] \\
&= \operatorname{tr} \left[C^{-1} \frac{\partial C}{\partial \alpha} \right] - \operatorname{tr} \left[Q^* \Lambda^{-1} Q \left(\frac{\partial Q^*}{\partial \alpha} \Lambda Q + Q^* \Lambda \frac{\partial Q}{\partial \alpha} \right) \right] \\
&= \operatorname{tr} \left[C^{-1} \frac{\partial C}{\partial \alpha} \right] - \operatorname{tr} \left[\frac{\partial Q^*}{\partial \alpha} Q + Q^* \frac{\partial Q}{\partial \alpha} \right] \\
&= \operatorname{tr} \left[C^{-1} \frac{\partial C}{\partial \alpha} \right]
\end{aligned}$$

which is the result stated. ■

Finally, we make use of a simple property of positive semidefinite matrices to prove the ***Cauchy–Schwartz inequality*** for vectors and for functions.

Result R22 (Cauchy–Schwartz inequality for vectors): Let $x,\ y \in \mathbf{C}^{m \times 1}$. Then

$$|x^* y|^2 \le \|x\|^2 \ \|y\|^2 \tag{A.5.9}$$

where $|\cdot|$ denotes the modulus of a possibly complex-valued number, and $\|\cdot\|$ denotes the Euclidean vector norm ($\|x\|^2 = x^* x$). Equality in (A.5.9) is achieved if and only if x is proportional to y.

Proof: The (2×2) matrix

$$\begin{bmatrix} \|x\|^2 & x^*y \\ y^*x & \|y\|^2 \end{bmatrix} = \begin{bmatrix} x^* \\ y^* \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix} \tag{A.5.10}$$

is positive semidefinite because condition (iii) in D11 is satisfied. It follows from condition (iv) in D11 that the determinant of the preceding matrix must be nonnegative; in other words,

$$\|x\|^2 \ \|y\|^2 - |x^*y|^2 \geq 0$$

which gives (A.5.9). Equality in (A.5.9) holds if and only if the determinant of (A.5.10) is equal to zero. The latter condition is equivalent to requiring that x be proportional to y. (*Cf.* D3: The columns of the matrix $[x \ y]$ will then be linearly dependent.) ■

Result R23 (Cauchy–Schwartz inequality for functions): Let $f(x)$ and $g(x)$ be two complex-valued functions defined for a real-valued argument x. Then, assuming that the needed integrals exist,

$$\left| \int_I f(x) g^*(x) dx \right|^2 \leq \left[\int_I |f(x)|^2 dx \right] \left[\int_I |g(x)|^2 dx \right]$$

where $I \subset \mathbf{R}$ is an integration interval. The inequality above becomes an equality if and only if $f(x)$ is proportional to $g(x)$ on I.

Proof: The matrix

$$\int_I \begin{bmatrix} f(x) \\ g(x) \end{bmatrix} \left[f^*(x) \ g^*(x) \right] dx$$

is seen to be positive semidefinite (because the integrand is a positive semidefinite matrix for every $x \in I$). Hence, the stated result follows from the type of argument used in the proof of Result R22. ■

A.6 MATRICES WITH SPECIAL STRUCTURE

In this section, we consider several types of matrices with a special structure, for which we prove some basic properties used in the text.

Definition D13: A matrix $A \in \mathbf{C}^{m \times n}$ is called ***Vandermonde*** if it has the structure

$$A = \begin{bmatrix} 1 & \cdots & 1 \\ z_1 & & z_n \\ \vdots & & \vdots \\ z_1^{m-1} & \cdots & z_n^{m-1} \end{bmatrix} \tag{A.6.1}$$

where $z_k \in \mathbf{C}$ are usually assumed to be distinct.

Result R24: Consider the matrix A in (A.6.1) with $z_k \neq z_p$ for $k, p = 1, \ldots, n$ and $k \neq p$. Also let $m \geq n$ and assume that $z_k \neq 0$ for all k. Then any n consecutive rows of A are linearly independent.

Proof: To prove the assertion, it is sufficient to show that the following $n \times n$ Vandermonde matrix is nonsingular:

$$\bar{A} = \begin{bmatrix} 1 & \cdots & 1 \\ z_1 & & z_n \\ \vdots & & \vdots \\ z_1^{n-1} & \cdots & z_n^{n-1} \end{bmatrix}$$

Let $\beta = [\beta_0 \cdots \beta_{n-1}]^* \neq 0$. The equation $\beta^* \bar{A} = 0$ is equivalent to

$$\beta_0 + \beta_1 z + \cdots + \beta_{n-1} z^{n-1} = 0 \quad \text{at} \quad z = z_k \quad (k = 1, \ldots, n) \tag{A.6.2}$$

However, (A.6.2) is impossible; an $(n-1)$-degree polynomial cannot have n zeroes. Hence, $\bar{A}$ has full rank. ■

Definition D14: A matrix $A \in \mathbf{C}^{m \times n}$ is called

- ***Toeplitz*** when A_{ij} is a function of $i - j$ only.
- ***Hankel*** when A_{ij} is a function of $i + j$ only.

Observe that a Toeplitz matrix has the same element along each diagonal, whereas a Hankel matrix has identical elements on each of the antidiagonals.

Result R25: The eigenvectors of a symmetric Toeplitz matrix $A \in \mathbf{R}^{m \times m}$ are either symmetric or skew symmetric. More precisely, if J denotes the exchange (or reversal) matrix

$$J = \begin{bmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{bmatrix}$$

and if x is an eigenvector of A, then either $x = Jx$ or $x = -Jx$.

Proof: By the property (3.5.3) proven in Section 3.5, A satisfies

$$AJx = JAx$$

or, equivalently,

$$(JAJ)x = Ax$$

for any $x \in \mathbf{C}^{m \times 1}$. Hence, we must have

$$JAJ = A \tag{A.6.3}$$

Let (λ, x) denote an eigenpair of A:

$$Ax = \lambda x \tag{A.6.4}$$

Combining (A.6.3) and (A.6.4) yields

$$\lambda Jx = JAx = J(JAJ)x = A(Jx) \tag{A.6.5}$$

Because the eigenvectors of a symmetric matrix are unique modulo multiplication by a scalar, it follows from (A.6.5) that

$$x = \alpha Jx \quad \text{for some } \alpha \in \mathbf{R}$$

As x (and, hence, Jx) must have unit norm, α must satisfy $\alpha^2 = 1 \Rightarrow \alpha = \pm 1$; thus, either $x = Jx$ (x is symmetric) or $x = -Jx$ (x is skew symmetric). ■

One can show that, for m even, the number of symmetric eigenvectors is $m/2$, as is the number of skew-symmetric eigenvectors; for m odd, the number of symmetric eigenvectors is $(m+1)/2$ and the number of skew-symmetric eigenvectors is $(m-1)/2$. (See [Cantoni and Butler 1976].)

For many additional results on Toeplitz matrices, the reader can consult [Iohvidov 1982; Böttcher and Silbermann 1983].

A.7 MATRIX INVERSION LEMMAS

The following formulas for *the inverse of a partitioned matrix* are used in the text:

Result R26: Let $A \in \mathbf{C}^{m\times m}$, $B \in \mathbf{C}^{n\times n}$, $C \in \mathbf{C}^{m\times n}$ and $D \in \mathbf{C}^{n\times m}$. Then, provided that the appropriate matrix inverses exist,

$$\begin{aligned}\begin{bmatrix} A & C \\ D & B \end{bmatrix}^{-1} &= \begin{bmatrix} I \\ 0 \end{bmatrix} A^{-1} \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} -A^{-1}C \\ I \end{bmatrix} (B - DA^{-1}C)^{-1} [-DA^{-1} \;\; I] \\ &= \begin{bmatrix} 0 \\ I \end{bmatrix} B^{-1} \begin{bmatrix} 0 & I \end{bmatrix} + \begin{bmatrix} I \\ -B^{-1}D \end{bmatrix} (A - CB^{-1}D)^{-1} [I \;\; -CB^{-1}]\end{aligned}$$

Proof: By direct verification. ■

By equating the top-left blocks in these two equations, we obtain the so-called *Matrix Inversion Lemma*:

Result R27 ***Matrix Inversion Lemma*:** Let A, B, C, and D be as in R26. Then, assuming that the matrix inverses exist,

$$(A - CB^{-1}D)^{-1} = A^{-1} + A^{-1}C(B - DA^{-1}C)^{-1}DA^{-1}$$

A.8 SYSTEMS OF LINEAR EQUATIONS

Let $A \in \mathbf{C}^{m\times n}$, $B \in \mathbf{C}^{m\times p}$, and $X \in \mathbf{C}^{n\times p}$. A general *system of linear equations* in X can be written as

$$AX = B \tag{A.8.1}$$

where A and B are given and X is the unknown matrix. The special case of (A.8.1) corresponding to $p = 1$ (for which X and B are vectors) is perhaps the most common one in applications. For the sake of generality, we consider the system (A.8.1) with $p \geq 1$. (The ESPRIT system of equations encountered in Section 4.7 is of the form of (A.8.1) with $p > 1$.) We say that (A.8.1) is ***exactly determined*** whenever $m = n$, ***overdetermined*** if $m > n$ and ***underdetermined*** if $m < n$. In the following discussion, we first address the case where (A.8.1) has an exact solution and then the cases where (A.8.1) cannot be exactly satisfied.

A.8.1 Consistent Systems

Result R28: The linear system (A.8.1) is ***consistent***, that is it admits an exact solution X, if and only if $\mathcal{R}(B) \subset \mathcal{R}(A)$ or equivalently

$$\operatorname{rank}([A\ B]) = \operatorname{rank}(A) \tag{A.8.2}$$

Proof: The result is readily shown by the use of rank and range properties. ■

Result R29: Let X_0 be a particular solution to (A.8.1). Then *the set of all solutions* to (A.8.1) is given by

$$X = X_0 + \Delta \tag{A.8.3}$$

where $\Delta \in \mathbf{C}^{n\times p}$ is any matrix whose columns are in $\mathcal{N}(A)$.

Proof: Obviously, (A.8.3) satisfies (A.8.1). To show that no solution outside the set (A.8.3) exists, let $\Omega \in \mathbf{C}^{n\times p}$ be a matrix whose columns do not all belong to $\mathcal{N}(A)$. Then $A\Omega \neq 0$ and

$$A(X_0 + \Delta + \Omega) = A\Omega + B \neq B$$

and hence, $X_0 + \Delta + \Omega$ is not a solution to $AX = B$. ■

Result R30: The system of linear equations (A.8.1) has a *unique solution* if and only if (A.8.2) holds and A has full column rank:

$$\operatorname{rank}(A) = n \leq m \tag{A.8.4}$$

Proof: The assertion follows from R28 and R29. ■

Next, let us assume that (A.8.1) is consistent but A does *not* satisfy (A.8.4) (hence, $\dim \mathcal{N}(A) \geq 1$). Then, according to R29, there are infinitely many solutions. In what follows, we obtain that (unique) solution X_0 that has *minimum norm.*

Result R31: Consider a linear system that satisfies the consistency condition in (A.8.2). Let A have rank $r \leq \min(m, n)$, and let

$$A = [\,\underbrace{U_1}_{r}\quad \underbrace{U_2}_{m-r}\,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \begin{matrix} \}\, r \\ \}\, n-r \end{matrix} = U_1 \Sigma_1 V_1^*$$

denote the SVD of A. (Here Σ_1 is nonsingular, *cf.* the discussion in Section A.4). Then

$$X_0 = V_1 \Sigma_1^{-1} U_1^* B \tag{A.8.5}$$

is the ***minimum-Frobenius-norm solution*** of (A.8.1), in the sense that

$$\|X_0\|^2 < \|X\|^2 \tag{A.8.6}$$

for any other solution $X \neq X_0$.

Proof: First, we verify that X_0 satisfies (A.8.1). We have

$$AX_0 = U_1 U_1^* B \tag{A.8.7}$$

In (A.8.7), $U_1 U_1^*$ is the orthogonal projector onto $\mathcal{R}(A)$ (*cf.* R17). Because B must belong to $\mathcal{R}(A)$ (see R28), we conclude that $U_1 U_1^* B = B$ and, hence, that X_0 is indeed a solution.

Next, we note that, according to R15,

$$\mathcal{N}(A) = \mathcal{R}(V_2)$$

Consequently, the general solution (A.8.3) can be written (*cf.* R29) as

$$X = X_0 + V_2 Q\ ; \qquad Q \in \mathbf{C}^{(n-r)\times p}$$

from which we obtain

$$\begin{aligned} \|X\|^2 &= \operatorname{tr}[(X_0^* + Q^* V_2^*)(X_0 + V_2 Q)] \\ &= \|X_0\|^2 + \|V_2 Q\|^2 > \|X_0\|^2 \quad \text{for } X \neq X_0 \end{aligned} \qquad \blacksquare$$

Definition D15: The matrix

$$A^\dagger \triangleq V_1 \Sigma_1^{-1} U_1^* \tag{A.8.8}$$

in (A.8.5) is the so-called ***Moore–Penrose pseudoinverse*** (or ***generalized inverse***) of A.

It can be shown that $A^\dagger$ is the unique solution to the following set of equations:

$$\begin{cases} AA^\dagger A = A \\ A^\dagger AA^\dagger = A^\dagger \\ A^\dagger A \text{ and } AA^\dagger \text{ are Hermitian} \end{cases}$$

Evidently, whenever A is square and nonsingular, we have $A^\dagger = A^{-1}$; this observation partly motivates the name "generalized inverse" (or "pseudoinverse") given to $A^\dagger$ in the general case.

The computation of a solution to (A.8.1), whenever one exists, is an important issue, which we address briefly in what follows. We begin by noting that, in the general case, there is no computer algorithm that can compute a solution to (A.8.1) *exactly* (i.e., without any numerical errors). In effect, the best we can hope for is to compute the exact solution to a slightly perturbed (fictitious) system of linear equations, given by

$$(A + \Delta_A)(X + \Delta_X) = B + \Delta_B \tag{A.8.9}$$

where Δ_A and Δ_B are small perturbation terms, the magnitude of which depends on the algorithm and the length of the computer word, and where Δ_X is the solution perturbation induced. An algorithm which, when applied to (A.8.1), provides a solution to (A.8.9) corresponding to perturbation terms (Δ_A, Δ_B) whose magnitude is of the order afforded by the "machine epsilon" is said to be *numerically stable*. Now, assuming that (A.8.1) has a unique solution (and, hence, that A satisfies (A.8.4)), one can show that the perturbations in A and B in (A.8.9) are retrieved in Δ_X multiplied by a proportionality factor given by

$$\text{cond}(A) = \sigma_1/\sigma_n \tag{A.8.10}$$

where σ_1 and σ_n are the largest and smallest singular values of A, respectively, and where "cond" is short for "condition." The system (A.8.1) is said to be *well conditioned* if the corresponding ratio (A.8.10) is "small" (that is, not much larger than 1). The ratio in (A.8.10) is called the *condition number* of the matrix A and is an important parameter of a given system of linear equations. Note, from the previous discussion, that even a numerically stable algorithm (i.e., one that induces quite small Δ_A and Δ_B) could yield an inaccurate solution X when applied to an ill-conditioned system of linear equations (i.e., a system with a very large cond(A)). For more details on the topic of this paragraph, including specific algorithms for solving linear systems, we refer the reader to [Stewart 1973; Golub and Van Loan 1989].

A.8.2 Inconsistent Systems

The systems of linear equations that appear in applications (such as those in this book) are quite often perturbed versions of a "nominal system," and usually they do *not* admit any exact solution. Such systems are said to be *inconsistent*, and frequently they are overdetermined and have a matrix A that has full column rank:

$$\text{rank}(A) = n \leq m \tag{A.8.11}$$

In what follows, we present two approaches to obtaining an approximate solution to an inconsistent system of linear equations

$$AX \simeq B \tag{A.8.12}$$

under the condition (A.8.11).

Definition D16: The ***least squares*** (LS) approximate solution to (A.8.12) is given by the minimizer X_{LS} of the following criterion:

$$\|AX - B\|^2$$

Equivalently, X_{LS} can be defined as follows: Obtain the minimal perturbation Δ_B that makes the system (A.8.12) consistent—that is,

$$\min \ \|\Delta_B\|^2 \quad \text{subject to} \quad AX = B + \Delta_B \tag{A.8.13}$$

Then derive X_{LS} by solving the system in (A.8.13) corresponding to the optimal perturbation Δ_B.

The LS solution introduced above can be obtained in several ways. A simple way is as follows:

Result R32: The LS solution to (A.8.12) is given by

$$X_{LS} = (A^*A)^{-1}A^*B \tag{A.8.14}$$

The inverse matrix in this equation exists, in view of (A.8.11).

Proof: The matrix B_0 that makes the system consistent and is of minimal distance (in the Frobenius-norm metric) from B is given by the orthogonal projection of (the columns of) B onto $\mathcal{R}(A)$:

$$B_0 = A(A^*A)^{-1}A^*B \tag{A.8.15}$$

To motivate (A.8.15) by using only the results proven so far in this appendix, we digress from the main proof and let U_1 denote an orthogonal basis of $\mathcal{R}(A)$. Then R17 implies that $B_0 = U_1 U_1^* B$. However, U_1 and A span the same subspace; hence, they must be related to one another by a nonsingular linear transformation: $U_1 = AQ$ ($|Q| \neq 0$). It follows from this observation that $U_1 U_1^* = AQQ^*A^*$ and also that $Q^*A^*AQ = I$, which lead to the following projector formula: $U_1 U_1^* = A(A^*A)^{-1}A^*$ (as used in (A.8.15)).

Next, we return to the proof of (A.8.14). The unique solution to

$$AX - B_0 = A[X - (A^*A)^{-1}A^*B]$$

is obviously (A.8.14), because $\dim \mathcal{N}(A) = 0$ by assumption. ■

The *LS* solution X_{LS} can be computed by means of the SVD of the $m \times n$ matrix A. The X_{LS} can, however, be obtained in a computationally more efficient way, as is briefly described below. Note that X_{LS} should *not* be computed by directly evaluating the formula in (A.8.14) as it stands. Briefly stated, the reason is as follows: Recall, from (A.8.10), that the condition number of A is given by

$$\text{cond}(A) = \sigma_1/\sigma_n \tag{A.8.16}$$

(Note that $\sigma_n \neq 0$ under (A.8.11).) When working directly on A, we find that the numerical errors made in the computation of X_{LS} can be shown to be proportional to (A.8.16). However, in (A.8.14), one would need to invert the matrix A^*A, whose condition number is

$$\text{cond}(A^*A) = \sigma_1^2/\sigma_n^2 = [\text{cond}(A)]^2 \tag{A.8.17}$$

Working with (A^*A) would therefore lead to much larger numerical errors during the computation of X_{LS} and is thus not advisable. The algorithm sketched in what follows derives X_{LS} by operating on A directly.

For any matrix A satisfying (A.8.11), there exist a unitary matrix $Q \in \mathbf{C}^{m\times m}$ and nonsingular upper triangular matrix $R \in \mathbf{C}^{n\times n}$ such that

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \triangleq [\, \underbrace{Q_1}_{n} \quad \underbrace{Q_2}_{m-n} \,] \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{A.8.18}$$

The previous factorization of A is called the *QR decomposition* (QRD). Inserting (A.8.18) into (A.8.14), we obtain

$$X_{LS} = R^{-1}Q_1^*B$$

Hence, once the QRD of A has been performed, X_{LS} can be obtained conveniently, as the solution of a triangular system of linear equations:

$$RX_{LS} = Q_1^*B \tag{A.8.19}$$

We note that the computation of the QRD is faster than that of the SVD (see, for example, [Stewart 1973; Golub and Van Loan 1989]).

The previous definition and derivation of X_{LS} make it clear that the LS approach derives an approximate solution to (A.8.12) by implicitly assuming that only the right-hand-side matrix, B, is perturbed. In applications, quite frequently *both A and B* are perturbed versions of some nominal (and unknown) matrices. In such cases, we may think of determining an approximate solution to (A.8.12) by explicitly recognizing the fact that neither A nor B is perturbation free. An approach based on this idea is described next (see, for example, [Van Huffel and Vandewalle 1991]).

Definition D17: The ***total least squares*** (TLS) approximate solution to (A.8.12) is defined as follows: First, derive the minimal perturbations Δ_A and Δ_B that make the system consistent—that is,

$$\min \ \|[\Delta_A \ \Delta_B]\|^2 \quad \text{subject to} \quad (A + \Delta_A)X = B + \Delta_B \tag{A.8.20}$$

Then, obtain X_{TLS} by solving the system in (A.8.20) corresponding to the optimal perturbations $(\Delta_A, \ \Delta_B)$.

A simple way to derive a more explicit formula for calculating the X_{TLS} is as follows:

Result R33: Let

$$[A \ B] = [\ \underbrace{\tilde{U}_1}_{n} \ \ \underbrace{\tilde{U}_2}_{m-n}\ ] \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \tilde{V}_1^* \\ \tilde{V}_2^* \end{bmatrix} \begin{matrix} \}\, n \\ \}\, p \end{matrix} \tag{A.8.21}$$

denote the SVD of the matrix $[A \ B]$. Furthermore, partition $\tilde{V}_2^*$ as

$$\tilde{V}_2^* = [\ \underbrace{\tilde{V}_{21}^*}_{n} \ \ \underbrace{\tilde{V}_{22}^*}_{p}\ ] \tag{A.8.22}$$

Then

$$X_{TLS} = -\tilde{V}_{21} \ \tilde{V}_{22}^{-1} \tag{A.8.23}$$

if $\tilde{V}_{22}^{-1}$ exists.

Proof: The optimization problem with constraints in (A.8.20) can be restated in the following way: Find the minimal perturbation $[\Delta_A \ \Delta_B]$ and the corresponding matrix X such that

$$\{\ [A \ B] + [\Delta_A \ \Delta_B]\ \} \begin{bmatrix} -X \\ I \end{bmatrix} = 0 \tag{A.8.24}$$

Because $\operatorname{rank} \begin{bmatrix} -X \\ I \end{bmatrix} = p$, $[\Delta_A \ \ \Delta_B]$ should be such that $\dim \mathcal{N}(\ [A \ \ B] + [\Delta_A \ \ \Delta_B]\ ) \geq p$ or, equivalently,

$$\operatorname{rank}(\ [A \ B] + [\Delta_A \ \Delta_B]\ ) \leq n \tag{A.8.25}$$

According to R18, the minimal-perturbation matrix $[\Delta_A \ \Delta_B]$ that achieves (A.8.25) is given by

$$[\Delta_A \ \Delta_B] = -\tilde{U}_2 \tilde{\Sigma}_2 \tilde{V}_2^* \tag{A.8.26}$$

Inserting (A.8.26) along with (A.8.21) into (A.8.24), we obtain the following matrix equation in X:

$$\tilde{U}_1 \tilde{\Sigma}_1 \tilde{V}_1^* \begin{bmatrix} -X \\ I \end{bmatrix} = 0$$

Equivalently we have

$$\tilde{V}_1^* \begin{bmatrix} -X \\ I \end{bmatrix} = 0 \tag{A.8.27}$$

Equation (A.8.27) implies that X must satisfy

$$\begin{bmatrix} -X \\ I \end{bmatrix} = \tilde{V}_2 Q = \begin{bmatrix} \tilde{V}_{21} \\ \tilde{V}_{22} \end{bmatrix} Q \tag{A.8.28}$$

for some nonsingular normalizing matrix Q. The expression (A.8.23) for X_{TLS} is readily obtained from (A.8.28). ■

The TLS solution in (A.8.23) is unique if and only if the singular values $\{\tilde{\sigma}_k\}$ of the matrix $[A\ B]$ are such that $\tilde{\sigma}_n > \tilde{\sigma}_{n+1}$ (this follows from R18). When $\tilde{V}_{22}$ is singular, the TLS solution does not exist; see [Van Huffel and Vandewalle 1991].

The computation of the X_{TLS} requires the SVD of the $m \times (n + p)$ matrix $[A\ B]$. The solution X_{TLS} can be rewritten in a slightly different form. Let $\tilde{V}_{11}$, $\tilde{V}_{12}$ be defined via the following partition of $\tilde{V}_1^*$:

$$\tilde{V}_1^* = [\,\underbrace{\tilde{V}_{11}}_{n} \quad \underbrace{\tilde{V}_{12}}_{p}\,]$$

The orthogonality condition $\tilde{V}_1^*\ \tilde{V}_2 = 0$ can be rewritten as

$$\tilde{V}_{11}\tilde{V}_{21} + \tilde{V}_{12}\tilde{V}_{22} = 0$$

which yields

$$X_{TLS} = -\tilde{V}_{21}\tilde{V}_{22}^{-1} = \tilde{V}_{11}^{-1}\tilde{V}_{12} \tag{A.8.29}$$

Because p is usually (much) smaller than n, the formula (A.8.23) for X_{TLS} can often be more efficient computationally than is (A.8.29). (For example, in the common case of $p = 1$, (A.8.23) does not require a matrix inversion, whereas (A.8.29) requires the calculation of an $n \times n$ matrix inverse.)

A.9 QUADRATIC MINIMIZATION

Several problems in this text require the solution to *quadratic minimization problems*. In this section, we make use of matrix-analysis techniques to derive two results: one on unconstrained minimization, the other on constrained minimization.

Result R34: Let A be an $(n \times n)$ Hermitian positive definite matrix, let X and B be $(n \times m)$ matrices, and let C be an $m \times m$ Hermitian matrix. Then the unique solution to the minimization problem

$$\min_X F(X), \quad F(X) = X^*AX + X^*B + B^*X + C \tag{A.9.1}$$

is given by

$$X_0 = -A^{-1}B, \qquad F(X_0) = C - B^*A^{-1}B \tag{A.9.2}$$

Here, the matrix minimization means $F(X_0) \le F(X)$ for every $X \neq X_0$; that is, $F(X) - F(X_0)$ is a positive semidefinite matrix.

Proof: Let $X = X_0 + \Delta$, where Δ is an arbitrary $(n \times m)$ complex matrix. Then

$$\begin{aligned} F(X) &= (-A^{-1}B + \Delta)^*A(-A^{-1}B + \Delta) + (-A^{-1}B + \Delta)^*B \\ &\quad + B^*(-A^{-1}B + \Delta) + C \\ &= \Delta^*A\Delta + F(X_0) \end{aligned} \tag{A.9.3}$$

Now, A is positive definite, so $\Delta^*A\Delta \ge 0$ for all nonzero Δ; thus, the minimum value of $F(X)$ is $F(X_0)$, and the result is proven. ■

We next present a result on linearly constrained quadratic minimization.

Result R35: Let A be an $(n \times n)$ Hermitian positive definite matrix, and let $X \in \mathbf{C}^{n\times m}$, $B \in \mathbf{C}^{n\times k}$, and $C \in \mathbf{C}^{m\times k}$. Assume that B has full column rank equal to k (hence $n \ge k$). Then the unique solution to the minimization problem

$$\min_X X^*AX \quad \text{subject to} \quad X^*B = C \tag{A.9.4}$$

is given by

$$X_0 = A^{-1}B(B^*A^{-1}B)^{-1}C^* \tag{A.9.5}$$

Proof: First, note that $(B^*A^{-1}B)^{-1}$ exists and that $X_0^*B = C$. Let $X = X_0 + \Delta$, where $\Delta \in \mathbf{C}^{n\times m}$ satisfies $\Delta^*B = 0$ (so that X also satisfies the constraint $X^*B = C$). Then

$$X^*AX = X_0^*AX_0 + X_0^*A\Delta + \Delta^*AX_0 + \Delta^*A\Delta \tag{A.9.6}$$

where the two middle terms are equal to zero:

$$\Delta^* A X_0 = \Delta^* B (B^* A^{-1} B)^{-1} C^* = 0$$

Hence,

$$X^* A X - X_0^* A X_0 = \Delta^* A \Delta \geq 0 \tag{A.9.7}$$

because A is positive definite. It follows from (A.9.7) that the minimizing X matrix is given by X_0. ■

A common special case of Result R35 is $m = k = 1$ (so X and B are both vectors) and $C = 1$. Then

$$X_0 = \frac{A^{-1} B}{B^* A^{-1} B}$$

Appendix B

Cramér–Rao Bound Tools

B.1 INTRODUCTION

In the text, we have kept the discussion of statistical aspects at a minimum for conciseness reasons. However, we have presented certain statistical tools and analyses that we have found useful to the understanding of the spectral analysis material discussed. In this appendix, we introduce some basic facts on an important statistical tool: the Cramér–Rao bound (abbreviated as CRB). We begin our discussion by explaining the importance of the CRB for *parametric spectral analysis*.

Let $\phi(\omega, \theta)$ denote a parametric spectral model, depending on a *real-valued* vector θ, and let $\phi(\omega, \hat{\theta})$ denote the spectral density estimated from N data samples. Assume that the estimate $\hat{\theta}$ of θ is *consistent*, so that the estimation error is small for large values of N. Then, by making use of a Taylor series expansion technique, we can write the estimation error $[\phi(\omega, \hat{\theta}) - \phi(\omega, \theta)]$ approximately as a linear function of $\hat{\theta} - \theta$, namely,

$$[\phi(\omega, \hat{\theta}) - \phi(\omega, \theta)] \simeq \psi^T(\omega, \theta)(\hat{\theta} - \theta) \tag{B.1.1}$$

where the symbol $\simeq$ denotes an asymptotically (in N) valid approximation and $\psi(\omega, \theta)$ is the gradient of $\phi(\omega, \theta)$ with respect to θ (evaluated at the true parameter values):

$$\psi(\omega, \theta) = \frac{\partial \phi(\omega, \theta)}{\partial \theta} \tag{B.1.2}$$

It follows from (B.1.1) that the mean squared error (MSE) of $\phi(\omega, \hat{\theta})$ is approximately given by

$$\text{MSE}[\phi(\omega, \hat{\theta})] \simeq \psi^T(\omega, \theta) P \psi(\omega, \theta) \qquad (\text{for } N \gg 1) \tag{B.1.3}$$

where

$$P = \mathrm{MSE}[\hat{\theta}] = E\left\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right\} \tag{B.1.4}$$

We see from (B.1.3) that the variance (or MSE) of the estimation errors in the spectral domain is linearly related to the variance (or MSE) of the parameter vector estimate $\hat{\theta}$, and so we can get an accurate spectral estimate only if we use an accurate parameter estimator. We start from this simple observation, which reduces the statistical analysis of $\phi(\omega, \hat{\theta})$ to the analysis of $\hat{\theta}$, to explain the importance of the CRB for the performance study of spectral analysis. Toward that end, we discuss several facts in the paragraphs that follow.

Assume that $\hat{\theta}$ is some *unbiased estimate* of θ (i.e., $E\{\hat{\theta}\} = \theta$), and let P denote the covariance matrix of $\hat{\theta}$ (*cf.* (B.1.4)):

$$P = E\left\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right\} \tag{B.1.5}$$

(Note that here we do not require that N be large.) Then, under quite general conditions, there is a matrix (which we denote by P_{cr}) such that

$$P \geq P_{cr} \tag{B.1.6}$$

in the sense that the difference $(P - P_{cr})$ is a positive semidefinite matrix. This is basically the celebrated Cramér–Rao bound result [Cramér 1946; Rao 1945]. We will derive the inequality (B.1.6) along with an expression for the CRB in the next section.

In view of (B.1.6), we may think of assessing the performance of a given estimation method by comparing its covariance matrix P with the CRB. Such a comparison would make perfect sense whenever the CRB is *achievable*—that is, whenever there exists an estimation method such that its P equals the CRB. Unfortunately, this is rarely the case for finite N. Additionally, *biased* estimators with MSEs smaller than the CRB can exist. (See, for example, [Stoica and Moses 1990; Stoica and Ottersten 1996].) Hence, in the *finite sample case* (particularly for small samples), comparing with the CRB does not really make much sense, because

(i) there might be no unbiased estimator that attains the CRB and, consequently, a large difference $(P - P_{cr})$ would not necessarily mean bad accuracy; and
(ii) the equality $P = P_{cr}$ does not necessarily mean that we have achieved the ultimate possible performance, because there might be biased estimators with lower MSE than the CRB.

In the *large sample case*, on the other hand, the utility of the CRB result for the type of parameter estimation problems addressed in the text is significant, as explained next.

Let $y \in \mathbf{R}^{N \times 1}$ denote the sample of available observations. Any estimate $\hat{\theta}$ of θ will be a function of y. We assume that both θ and y are *real valued*. Working with real θ and y vectors appears to be the most convenient way when discussing the CRB theory, even when the original parameters and measurements are complex-valued. (If the parameters and measurements are complex-valued, θ and y are obtained by concatenating the real and imaginary parts of the complex parameter and data vectors, respectively.) We also assume that the probability density of y, which we denote by $p(y, \theta)$, is a differentiable function of θ. An important general method for parameter estimation consists of maximizing $p(y, \theta)$ with respect to θ:

$$\hat{\theta} = \arg\max_{\theta} p(y, \theta) \tag{B.1.7}$$

The $p(y, \theta)$ in (B.1.7) with y fixed and θ variable is called the *likelihood function*, and $\hat{\theta}$ is called the *maximum likelihood* (ML) *estimate* of θ. Under regularity conditions, the ML estimate (MLE) is *consistent* (i.e., $\lim_{N\to\infty} \hat{\theta} = \theta$ stochastically), and its covariance matrix approaches the CRB as N increases:

$$P \simeq P_{cr} \qquad \text{for a MLE with } N \gg 1 \tag{B.1.8}$$

The aforementioned regularity conditions basically amount to requiring that the number of free parameters not increase with N, which is true for all but one of the parametric spectral estimation problems discussed in the text. The array processing problem of Chapter 6 does not satisfy the previous requirement when the signal snapshots are assumed to be unknown deterministic variables; in such a case, the number of unknown parameters grows without bound as N increases, and the equality in (B.1.8) does not hold; see [Stoica and Nehorai 1989a; Stoica and Nehorai 1990] and also Section B.6.

In summary, then, in *large samples*, the ML method attains the ultimate performance corresponding to the CRB, under rather general conditions. Furthermore, there are no other known *practical methods* that can provide consistent estimates of θ with lower variance than the CRB.[1] Hence, the ML method can be said to be asymptotically a *statistically efficient practical estimation approach*. The accuracy achieved by any other estimation method can therefore be assessed *by comparing the (large-sample) covariance matrix of that method with the CRB*, which approximately equals the covariance matrix of the MLE in large samples (*cf.* (B.1.8)). This performance-comparison ability is one of the most important uses of the CRB.

With reference to the spectral estimation problem, it follows from (B.1.3) and the previous observation that we can assess the performance of a given spectral estimator by comparing its large sample MSE values with

$$\boxed{\psi^T(\omega, \theta)[P_{cr}]\psi(\omega, \theta)} \tag{B.1.9}$$

The MSE values can be obtained either by the Monte Carlo simulation of a typical scenario representative of the problem of interest or by using analytical MSE formulas whenever they are available. In this book, we have emphasized the former, more pragmatic way of finding the MSE of a given spectral estimator.

[1] Consistent estimation methods whose asymptotic variance is lower than the CRB, at certain points in the parameter set, do exist! However, such methods (which are called "asymptotically statistically superefficient") have little practical relevance (they are mainly of a theoretical interest); see, for example, [Stoica and Ottersten 1996].

Remark: The CRB formula (B.1.9) for parametric (or model-based) spectral analysis holds in the case where the model order (i.e., the dimension of θ) is equal to the "true order." Of course, in any practical spectral analysis exercise using the parametric approach, we will have to estimate n, the model order, in addition to θ, the (real-valued) model parameters. The need for order estimation is a distinctive feature and an additional complication of parametric spectral analysis, as compared with nonparametric spectral analysis.

There are several available rules for order selection; see Appendix C. For most of these rules, the probability of underestimating the true order approaches zero as N increases (if that is not the case, then the estimated spectrum could be heavily biased). The probability of overestimating the true order, on the other hand, may be nonzero even when $N \to \infty$. Let $\hat{n}$ denote the estimated order, n_0 the true order, and $p_n = \Pr(\hat{n} = n)$ for $N \to \infty$. Assume that $p_n = 0$ for $n < n_0$ and that the CRB formula (B.1.9) holds for any $n \geq n_0$ (which is a relatively mild restriction). Then it can be shown (see [Sando, Mitra, and Stoica 2002] and the references therein) that, whenever n is estimated along with θ, the formula (B.1.9) should be replaced with its average over the distribution of order estimates:

$$\boxed{\sum_{n=n_0}^{n_{MAX}} p_n \psi_n^T(\omega, \theta_n)[P_{cr,n}]\psi_n(\omega, \theta_n)} \tag{B.1.10}$$

Here we have emphasized by notation the dependence of ψ, θ, and P_{cr} on the model order n, and n_{MAX} denotes the maximum order value considered in the order-selection rule. The set of probabilities $\{p_n\}$ associated with various order-estimation rules is tabulated e.g., in [McQuarrie and Tsai 1998]. As expected, it can be proven that the spectral CRB in (B.1.10) increases (for each ω) with increasing n_{MAX} (see [Sando, Mitra, and Stoica 2002]). This increase of the spectral-estimation error is the price paid for not knowing the true model order. ■

B.2 THE CRB FOR GENERAL DISTRIBUTIONS

Result R36: *(Cramér–Rao Bound)* Consider the likelihood function $p(y, \theta)$, introduced in the previous section, and define

$$\boxed{P_{cr} = \left(E\left\{ \left[\frac{\partial \ln p(y, \theta)}{\partial \theta}\right] \left[\frac{\partial \ln p(y, \theta)}{\partial \theta}\right]^T \right\} \right)^{-1}} \tag{B.2.1}$$

where the inverse is assumed to exist. Then

$$\boxed{P \geq P_{cr}} \tag{B.2.2}$$

holds for any unbiased estimate of θ. Furthermore, the CRB matrix can alternatively be expressed as

$$\boxed{P_{cr} = -\left(E\left\{\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\,\partial\theta^T}\right\}\right)^{-1}} \tag{B.2.3}$$

Proof: As $p(y,\theta)$ is a probability density function,

$$\int p(y,\theta)dy = 1 \tag{B.2.4}$$

where the integration is over $\mathbf{R}^N$. The assumption that $\hat{\theta}$ is an unbiased estimate implies

$$\int \hat{\theta} p(y,\theta)dy = \theta \tag{B.2.5}$$

Differentiation of (B.2.4) and (B.2.5) with respect to θ yields, under regularity conditions,

$$\int \frac{\partial p(y,\theta)}{\partial\theta}dy = \int \frac{\partial \ln p(y,\theta)}{\partial\theta}p(y,\theta)dy = E\left\{\frac{\partial \ln p(y,\theta)}{\partial\theta}\right\} = 0 \tag{B.2.6}$$

and

$$\int \hat{\theta}\frac{\partial p(y,\theta)}{\partial\theta}dy = \int \hat{\theta}\frac{\partial \ln p(y,\theta)}{\partial\theta}p(y,\theta)dy = E\left\{\hat{\theta}\frac{\partial \ln p(y,\theta)}{\partial\theta}\right\} = I \tag{B.2.7}$$

It follows from (B.2.6) and (B.2.7) that

$$E\left\{(\hat{\theta}-\theta)\frac{\partial \ln p(y,\theta)}{\partial\theta}\right\} = I \tag{B.2.8}$$

Next note that the matrix

$$E\left\{\left[\begin{array}{c}(\hat{\theta}-\theta)\\ \dfrac{\partial \ln p(y,\theta)}{\partial\theta}\end{array}\right]\left[\begin{array}{cc}(\hat{\theta}-\theta)^T & \left(\dfrac{\partial \ln p(y,\theta)}{\partial\theta}\right)^T\end{array}\right]\right\} = \left[\begin{array}{cc}P & I\\ I & P_{cr}^{-1}\end{array}\right] \tag{B.2.9}$$

is, by construction, positive semidefinite. (To obtain the equality in (B.2.9), we used (B.2.8).) This observation implies (B.2.2) (see Result R20 in Appendix A).

Next, we prove the equality in (B.2.3). Differentiation of (B.2.6) gives

$$\int \frac{\partial^2 \ln p(y,\theta)}{\partial\theta\,\partial\theta^T}p(y,\theta)dy + \int \left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]\left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]^T p(y,\theta)dy = 0$$

or, equivalently,

$$E\left\{\left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]\left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]^T\right\} = -E\left\{\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\,\partial\theta^T}\right\}$$

which is precisely what we had to prove. ■

The matrix

$$\begin{aligned} J &= E\left\{\left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]\left[\frac{\partial \ln p(y,\theta)}{\partial\theta}\right]^T\right\} \\ &= -E\left\{\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\,\partial\theta^T}\right\}, \end{aligned} \tag{B.2.10}$$

the inverse of which appears in the CRB formula (B.2.1) (or (B.2.3)), is called the (Fisher) *information matrix* [Fisher 1922].

B.3 THE CRB FOR GAUSSIAN DISTRIBUTIONS

The CRB matrix in (B.2.1) depends implicitly on the data properties via the probability density function $p(y,\theta)$. To obtain a more explicit expression for the CRB, we should specify the data distribution. A particularly convenient CRB formula is obtained if the data vector is assumed to be Gaussian distributed—that is,

$$p(y,\theta) = \frac{1}{(2\pi)^{N/2}|C|^{1/2}} e^{-(y-\mu)^T C^{-1}(y-\mu)/2} \tag{B.3.1}$$

where μ and C are, respectively, the mean and the covariance matrix of y and C is assumed to be invertible. In the case of (B.3.1), the log-likelihood function that appears in (B.2.1) is given by

$$\ln p(y,\theta) = -\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|C| - \frac{1}{2}(y-\mu)^T C^{-1}(y-\mu) \tag{B.3.2}$$

Result R37: The CRB matrix corresponding to the Gaussian data distribution in (B.3.1) is given (elementwise) by

$$\boxed{[P_{cr}^{-1}]_{ij} = \frac{1}{2}\,\mathrm{tr}\left[C^{-1}C_i'C^{-1}C_j'\right] + \left[\mu_i'^T C^{-1}\mu_j'\right]} \tag{B.3.3}$$

where C_i' denotes the derivative of C with respect to the ith element of θ (and similarly for μ_i').

Proof: By using Result R21 and the notational foregoing convention for the first-order and second-order derivatives, we obtain

$$
\begin{aligned}
2[\ln p(y,\theta)]''_{ij} &= \frac{\partial}{\partial\theta_i}\Big\{-\operatorname{tr}\left[C^{-1}C'_j\right] + 2\mu_j'^T C^{-1}(y-\mu) \\
&\qquad +(y-\mu)^T C^{-1}C'_j C^{-1}(y-\mu)\Big\} \\
&= \operatorname{tr}\left[C^{-1}C'_i C^{-1}C'_j\right] - \operatorname{tr}\left[C^{-1}C''_{ij}\right] \\
&\quad +2\left\{\left[\mu_j'^T C^{-1}\right]'_i (y-\mu) - \mu_j'^T C^{-1}\mu'_i\right\} \\
&\quad -2\mu_i'^T C^{-1}C'_j C^{-1}(y-\mu) \\
&\quad +\operatorname{tr}\Big\{(y-\mu)(y-\mu)^T \\
&\qquad \cdot\left[-C^{-1}C'_i C^{-1}C'_j C^{-1} + C^{-1}C''_{ij}C^{-1} - C^{-1}C'_j C^{-1}C'_i C^{-1}\right]\Big\}
\end{aligned}
$$

Taking the expectation of both sides of the preceding equation yields

$$
\begin{aligned}
2\left[P_{cr}^{-1}\right]_{ij} &= -\operatorname{tr}\left[C^{-1}C'_i C^{-1}C'_j\right] + \operatorname{tr}\left[C^{-1}C''_{ij}\right] + 2\mu_i'^T C^{-1}\mu'_j \\
&\quad +\operatorname{tr}\left[C^{-1}C'_i C^{-1}C'_j\right] - \operatorname{tr}\left[C^{-1}C''_{ij}\right] + \operatorname{tr}\left[C^{-1}C'_i C^{-1}C'_j\right] \\
&= \operatorname{tr}\left[C^{-1}C'_i C^{-1}C'_j\right] + 2\mu_i'^T C^{-1}\mu'_j
\end{aligned}
$$

which concludes the proof. ■

The CRB expression in (B.3.3) is sometimes referred to as the *Slepian–Bangs formula*. (The second term in (B.3.3) is due to Slepian [SLEPIAN 1954] and the first to Bangs [BANGS 1971].)

Next, we specialize the CRB formula (B.3.3) to a particular type of Gaussian distribution. Let $N = 2\bar{N}$ (hence, N is assumed to be even). Partition the vector y as

$$
y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{matrix} \}\bar{N} \\ \}\bar{N} \end{matrix} \tag{B.3.4}
$$

Accordingly, partition μ and C as

$$
\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \tag{B.3.5}
$$

and

$$
C = \begin{bmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{bmatrix} \tag{B.3.6}
$$

The vector y is said to have a *circular* (or *circularly symmetric*) *Gaussian distribution* if

$$\boxed{C_{11} = C_{22}} \tag{B.3.7}$$

$$\boxed{C_{12}^T = -C_{12}} \tag{B.3.8}$$

Let

$$\mathbf{y} \overset{\triangle}{=} y_1 + iy_2 \tag{B.3.9}$$

and

$$\boldsymbol{\mu} = \mu_1 + i\mu_2 \tag{B.3.10}$$

We also say that the *complex-valued random vector* **y** *has a circular Gaussian distribution* whenever the conditions (B.3.7) and (B.3.8) are satisfied. It is a straightforward exercise to verify that the aforementioned conditions can be more compactly written as:

$$\boxed{E\left\{(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})^T\right\} = 0} \tag{B.3.11}$$

Both the Fourier transform (see Chapter 2) and the complex demodulation operation (see Chapter 6) often lead to signals satisfying (B.3.11) (see, e.g., [BRILLINGER 1981]). Hence, the *circularity* is a relatively common property of Gaussian random signals encountered in spectral analysis problems.

Remark: If a random vector **y** satisfies the "circularity condition" (B.3.11), then it is readily verified that **y** and $\mathbf{y}e^{iz}$ have the same second-order properties for every constant z in $[-\pi, \pi]$. Hence, the second-order properties of **y** do not change if its generic element $\mathbf{y}_k$ is replaced by any other value, $\mathbf{y}_k e^{iz}$, on the *circle* with radius $|\mathbf{y}_k|$ (recall that z is nonrandom and it does not depend on k). This observation provides a motivation for the name "circularly symmetric" given to such a random vector **y**. ■

Let

$$\Gamma = E\left\{(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})^*\right\} \tag{B.3.12}$$

For circular Gaussian random vectors y (or **y**), the CRB formula (B.3.3) can be rewritten in a compact form as a function of Γ and $\boldsymbol{\mu}$. (Note that the dimensions of Γ and $\boldsymbol{\mu}$ are half the dimensions of C and μ appearing in (B.3.3).) In order to show how this can be done, we need some preparations.

Let

$$\bar{C} = C_{11} = C_{22} \tag{B.3.13}$$

$$\tilde{C} = C_{12}^T = -C_{12} \tag{B.3.14}$$

Hence,

$$C = \begin{bmatrix} \bar{C} & -\tilde{C} \\ \tilde{C} & \bar{C} \end{bmatrix} \tag{B.3.15}$$

and

$$\Gamma = 2(\bar{C} + i\tilde{C}) \tag{B.3.16}$$

To any complex-valued matrix $\mathcal{C} = \bar{C} + i\tilde{C}$ we associate a real-valued matrix C as defined in (B.3.15), and vice versa. It is a simple exercise to verify that, if

$$\mathcal{A} = \mathcal{B}\mathcal{C} \Longleftrightarrow \bar{A} + i\tilde{A} = (\bar{B} + i\tilde{B})(\bar{C} + i\tilde{C}) \tag{B.3.17}$$

then the real-valued matrix associated with $\mathcal{A}$ is given by

$$A = BC \Longleftrightarrow \begin{bmatrix} \bar{A} & -\tilde{A} \\ \tilde{A} & \bar{A} \end{bmatrix} = \begin{bmatrix} \bar{B} & -\tilde{B} \\ \tilde{B} & \bar{B} \end{bmatrix} \begin{bmatrix} \bar{C} & -\tilde{C} \\ \tilde{C} & \bar{C} \end{bmatrix} \tag{B.3.18}$$

In particular, it follows from (B.3.17) and (B.3.18) with $A = I$ (and hence $\mathcal{A} = I$) that the matrices C^{-1} and $\mathcal{C}^{-1}$ form a real-complex pair as just defined.

We deduce from the results previously derived that the matrix in the first term of (B.3.3),

$$D = C^{-1}C_i'C^{-1}C_j' \tag{B.3.19}$$

is associated with

$$\mathcal{D} = \mathcal{C}^{-1}\mathcal{C}_i'\mathcal{C}^{-1}\mathcal{C}_j' = \Gamma^{-1}\Gamma_i'\Gamma^{-1}\Gamma_j' \tag{B.3.20}$$

Furthermore, we have

$$\frac{1}{2}\operatorname{tr}(D) = \operatorname{tr}(\bar{D}) = \operatorname{tr}(\mathcal{D}) \tag{B.3.21}$$

The second equality in (B.3.21) follows from the fact that $\mathcal{C}$ is Hermitian, and hence

$$\operatorname{tr}(\mathcal{D}^*) = \operatorname{tr}(\mathcal{C}_j'\mathcal{C}^{-1}\mathcal{C}_i'\mathcal{C}^{-1}) = \operatorname{tr}(\mathcal{C}^{-1}\mathcal{C}_i'\mathcal{C}^{-1}\mathcal{C}_j') = \operatorname{tr}(\mathcal{D})$$

which in turn implies that $\operatorname{tr}(\tilde{D}) = 0$ and therefore that $\operatorname{tr}(\mathcal{D}) = \operatorname{tr}(\bar{D})$. Combining (B.3.20) and (B.3.21) shows that the first term in (B.3.3) can be rewritten as

$$\operatorname{tr}(\Gamma^{-1}\Gamma_i'\Gamma^{-1}\Gamma_j') \tag{B.3.22}$$

Next, we consider the second term in (B.3.3). Let

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{and} \qquad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

be two arbitrary vectors partitioned similarly to μ, and let $\mathbf{x} = x_1 + ix_2$ and $\mathbf{z} = z_1 + iz_2$. A straightforward calculation shows that

$$\begin{aligned} x^T A z &= x_1^T \bar{A} z_1 + x_2^T \bar{A} z_2 + x_2^T \tilde{A} z_1 - x_1^T \tilde{A} z_2 \\ &= \operatorname{Re}\left\{\mathbf{x}^* \mathcal{A} \mathbf{z}\right\} \end{aligned} \tag{B.3.23}$$

Hence,

$$\begin{aligned} \mu_i'^T C^{-1} \mu_j' &= \operatorname{Re}\left\{\boldsymbol{\mu}_i'^* \mathcal{C}^{-1} \boldsymbol{\mu}_j'\right\} \\ &= 2 \operatorname{Re}\left\{\boldsymbol{\mu}_i'^* \Gamma^{-1} \boldsymbol{\mu}_j'\right\} \end{aligned} \tag{B.3.24}$$

Insertion of (B.3.22) and (B.3.24) into (B.3.3) yields the following CRB formula, which holds in the case of *circularly Gaussian-distributed data vectors y* (or $\mathbf{y}$):

$$\boxed{[P_{cr}^{-1}]_{ij} = \operatorname{tr}\left[\Gamma^{-1}\Gamma_i'\Gamma^{-1}\Gamma_j'\right] + 2\operatorname{Re}\left[\boldsymbol{\mu}_i'^*\Gamma^{-1}\boldsymbol{\mu}_j'\right]} \tag{B.3.25}$$

The importance of the Gaussian CRB formulas lies not only in the fact that Gaussian data are rather frequently encountered in applications, but also in a more subtle aspect, explained in what follows. Briefly stated, the second reason for the importance of the CRB formulas derived in this section is that

> Under rather general conditions and (at least) in large samples, the Gaussian CRB is the largest of all CRB matrices corresponding to different congruous distributions of the data sample.[2]

(B.3.26)

To motivate the previous assertion, consider the ML estimate of θ derived under the Gaussian data hypothesis, which we denote by $\hat{\theta}_G$. According to the discussion around equation (B.1.8), the large sample covariance matrix of $\hat{\theta}$ equals P_{cr}^G—as with $\hat{\theta}_G$, we use an index G to denote the CRB matrix in the Gaussian-hypothesis case. Now, under rather general conditions, the large sample properties of the Gaussian ML estimator are independent of the data distribution. (See, for example, [Söderström and Stoica 1989].) In other words, the large sample covariance matrix of $\hat{\theta}_G$ is equal to P_{cr}^G for many data distributions other than the Gaussian one. This observation, along with the general CRB inequality, implies that

$$P_{cr}^G \geq P_{cr} \tag{B.3.27}$$

where the right-hand side is the CRB matrix corresponding to the data distribution at hand.

[2] A meaningful comparison of the CRBs under two different data distributions requires that the hypothesized distributional models not contain conflicting assumptions. In particular, when one of the two distributions is the Gaussian, the mean and covariance matrix should be the same for both distributions.

The inequality (B.3.27) (or, equivalently, the assertion (B.3.26)) shows that a method whose covariance matrix is much larger than P_{cr}^{G} cannot be a good estimation method. As a matter of fact, the "asymptotic properties" of most existing parameter estimation methods do not depend on the data distribution. This means that P_{cr}^{G} is a lower bound for the covariance matrices of a large class of estimation methods, regardless of the data distribution. On the other hand, the inequality (B.3.27) also shows that, for non-Gaussian data, it should be possible to beat the Gaussian CRB (for instance, by exploiting higher order moments of the data, beyond the first- and second-order moments used in the Gaussian ML method). However, general estimation methods with covariance matrices uniformly smaller than P_{cr}^{G} are yet to be discovered. In summary, comparing against the P_{cr}^{G} makes sense in most parameter estimation exercises.

In what follows, we briefly consider the application of the general Gaussian CRB formulas derived in this section to the three main parameter estimation problems treated in the text.

B.4 THE CRB FOR LINE SPECTRA

As explained in Chapter 4, the estimation of line spectra is basically a parameter estimation problem. The corresponding parameter vector is

$$\theta = \left[\alpha_1 \ \ldots \ \alpha_n, \ \varphi_1 \ \ldots \ \varphi_n, \ \omega_1 \ \ldots \ \omega_n, \ \sigma^2\right]^T \tag{B.4.1}$$

and the data vector is

$$\mathbf{y} = \left[y(1) \cdots y(N)\right]^T \tag{B.4.2}$$

or, in real valued form,

$$y = \left[\ \mathrm{Re}[y(1)] \cdots \mathrm{Re}[y(N)] \ \ \mathrm{Im}[y(1)] \cdots \mathrm{Im}[y(N)]\ \right]^T \tag{B.4.3}$$

When $\{\varphi_k\}$ are assumed to be random variables uniformly distributed on $[0, 2\pi]$ (whereas $\{\alpha_k\}$ and $\{\omega_k\}$ are deterministic constants), the distribution of $\mathbf{y}$ is *not* Gaussian and, hence, neither of the CRB formulas of the previous section is usable. To overcome this difficulty, it is customary to consider the distribution of $\mathbf{y}$ *conditioned on* $\{\varphi_k\}$ (i.e., for $\{\varphi_k\}$ fixed). This distribution is circular Gaussian, under the assumption that the (white) noise is circularly Gaussian distributed, with the following mean and covariance matrix:

$$\boldsymbol{\mu} = E\left\{\mathbf{y}\right\} = \begin{bmatrix} 1 & \cdots & 1 \\ e^{i\omega_1} & \cdots & e^{i\omega_n} \\ \vdots & & \vdots \\ e^{i(N-1)\omega_1} & \cdots & e^{i(N-1)\omega_n} \end{bmatrix} \begin{bmatrix} \alpha_1 e^{i\varphi_1} \\ \vdots \\ \alpha_n e^{i\varphi_n} \end{bmatrix} \tag{B.4.4}$$

$$\Gamma = E\left\{(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})^*\right\} = \sigma^2 I \tag{B.4.5}$$

The differentiation of (B.4.4) and (B.4.5) with respect to the elements of the parameter vector θ can be done easily; we leave the details of this differentiation operation as an exercise to the reader. Hence, we can readily obtain all ingredients required to evaluate the CRB matrix in equation (B.3.25). If the distribution of $\mathbf{y}$ (or y) is Gaussian but not circular, we need additional parameters, besides σ^2, to characterize the matrix $E\left\{(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^T\right\}$. Once these parameters are introduced, the use of formula (B.3.3) to obtain the CRB is straightforward.

In Section 4.3, we gave a simple formula for the block of the CRB matrix corresponding to the frequency estimates $\{\hat{\omega}_k\}$. That formula holds asymptotically, as N increases. For finite values of N, it is a good approximation of the exact CRB whenever the minimum frequency separation is larger than $1/N$ [Stoica, Moses, Friedlander, and Söderström 1989]. In any case, the approximate (large-sample) CRB formula given in Section 4.3 is computationally much simpler to implement than the exact CRB.

The computation and properties of the CRB for line-spectral models are discussed in great detail in [Ghogho and Swami 1999]. In particular, a modified lower bound on the variance of any unbiased estimates of $\{\alpha_k\}$ and $\{\omega_k\}$ is derived for the case in which $\{\varphi_k\}$ are independent random variables uniformly distributed on $[0, 2\pi]$. That bound, which was obtained by using the so-called posterior CRB introduced in [Van Trees 1968] (as indicated above, the standard CRB does not apply to such a case), has an expression that is quite similar to the large-sample CRB given in [Stoica, Moses, Friedlander, and Söderström 1989] (see Section 4.3 for the large-sample CRB for $\{\hat{\omega}_k\}$). The paper [Ghogho and Swami 1999] also discusses the derivation of the CRB in the case of non-Gaussian noise distributions. The extension of the asymptotic CRB formula in Section 4.3 to the case of colored noise can be found in [Stoica, Jakobsson, and Li 1997].

B.5 THE CRB FOR RATIONAL SPECTRA

For rational (or ARMA) spectra, the Cramér–Rao lower bound on the variance of any consistently estimated spectrum is given by (B.1.9). The CRB matrix for the parameter-vector estimate, which appears in (B.1.9), can be evaluated as outlined in what follows.

In the case of ARMA spectral models, the parameter vector consists of the white-noise power σ^2 and the polynomial coefficients $\{a_k, b_k\}$. We arrange the ARMA coefficients in the following real-valued vector:

$$\theta = [\mathrm{Re}(a_1)\cdots\mathrm{Re}(a_n)\ \ \mathrm{Re}(b_1)\cdots\mathrm{Re}(b_m)\,\mathrm{Im}(a_1)\cdots\mathrm{Im}(a_n)\ \ \mathrm{Im}(b_1)\cdots\mathrm{Im}(b_m)]^T$$

The data vector is defined as in equations (B.4.2) or (B.4.3) and has zero mean ($\mu = 0$). The calculation of the covariance matrix of the data vector reduces to the calculation of ARMA covariances—that is,

$$r(k) = \sigma^2 E\left\{\left[\frac{B(z)}{A(z)}w(t)\right]\left[\frac{B(z)}{A(z)}w(t-k)\right]^*\right\}$$

where the white-noise sequence $\{w(t)\}$ is normalized in such a way that its variance is 1. Methods for computation of $\{r_k\}$ (for given values of σ^2 and θ) were outlined in Exercises C1.12 and 3.2.

The method in Exercise C1.12 should perform reasonably well as long as the zeroes of $A(z)$ are not too close to the unit circle. If the zeroes of $A(z)$ are close to the unit circle, it is advisable to use the method in Exercise 3.2 or in [Kinkel, Perl, Scharf, and Stubberud 1979; Demeure and Mullis 1989].

The calculation of the derivatives of $\{r(k)\}$ with respect to σ^2 and the elements of θ, which appear in the CRB formulas (B.3.3) or (B.3.25), can also be reduced to ARMA (cross)covariance computation. To see this, let α and γ be the real parts of a_p and b_p, respectively. Then

$$\begin{aligned}\frac{\partial r(k)}{\partial \alpha} = -\sigma^2 E\Bigg\{&\left[\frac{B(z)}{A^2(z)}w(t-p)\right]\left[\frac{B(z)}{A(z)}w(t-k)\right]^* \\ &+\left[\frac{B(z)}{A(z)}w(t)\right]\left[\frac{B(z)}{A^2(z)}w(t-k-p)\right]^*\Bigg\}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial r(k)}{\partial \gamma} = \sigma^2 E\Bigg\{&\left[\frac{1}{A(z)}w(t-p)\right]\left[\frac{B(z)}{A(z)}w(t-k)\right]^* \\ &+\left[\frac{B(z)}{A(z)}w(t)\right]\left[\frac{1}{A(z)}w(t-k-p)\right]^*\Bigg\}\end{aligned}$$

The derivatives of $r(k)$ with respect to the imaginary parts of a_p and b_p can be similarly obtained. The differentiation of $r(k)$ with respect to σ^2 is immediate. Hence, by making use of an algorithm for ARMA cross-covariance calculation (similar to the ones for autocovariance calculation in Exercises C1.12 and 3.2) we can readily obtain all the ingredients needed to evaluate the CRB matrix in equation (B.3.3) or (B.3.25).

As in the case of line spectra, for relatively large values of N (e.g., on the order of hundreds), the use of the exact CRB formula for rational spectra could be burdensome computationally (given the need to multiply and invert matrices of large dimensions). In such large-sample cases, we might want to use an asymptotically valid approximation of the exact CRB, such as the one developed in [Söderström and Stoica 1989]. Below we present such an approximate (large-sample) CRB formula for ARMA parameter estimates.

Let

$$\Lambda = E\left\{\begin{bmatrix}\mathrm{Re}[e(t)] \\ \mathrm{Im}[e(t)]\end{bmatrix}\begin{bmatrix}\mathrm{Re}[e(t)] & \mathrm{Im}[e(t)]\end{bmatrix}\right\} \tag{B.5.1}$$

Typically, the real and imaginary parts of the complex-valued white-noise sequence $\{e(t)\}$ are assumed to be mutually uncorrelated and have the same variance $\sigma^2/2$. In such a case, we have $\Lambda = (\sigma^2/2)I$. However, this assumption is not necessary for the result discussed below to hold; hence, we do not impose it. (In other words, Λ in (B.5.1) is constrained only to be a positive definite matrix.) We should also remark that, for the sake of simplicity, we assumed that the ARMA signal under discussion is scalar. Nevertheless, the extension of the discussion that follows to multivariate ARMA signals is immediate. Finally, note that, for real-valued signals, the imaginary parts in (B.5.1) (and in equation (B.5.2)) should be omitted.

The real-valued white noise vector in (B.5.1) satisfies the equation

$$\underbrace{\begin{bmatrix} \mathrm{Re}[e(t)] \\ \\ \mathrm{Im}[e(t)] \end{bmatrix}}_{\varepsilon(t)} = \underbrace{\begin{bmatrix} \mathrm{Re}\left[\dfrac{A(z)}{B(z)}\right] & -\,\mathrm{Im}\left[\dfrac{A(z)}{B(z)}\right] \\ \mathrm{Im}\left[\dfrac{A(z)}{B(z)}\right] & \mathrm{Re}\left[\dfrac{A(z)}{B(z)}\right] \end{bmatrix}}_{H(z)} \underbrace{\begin{bmatrix} \mathrm{Re}[y(t)] \\ \\ \mathrm{Im}[y(t)] \end{bmatrix}}_{v(t)} \tag{B.5.2}$$

where z^{-1} is to be treated as the unit delay operator (not as a complex variable). As the coefficients of the polynomials $A(z)$ and $B(z)$ in $H(z)$ above are the unknowns in our estimation problem, we can rewrite (B.5.2) in the following form to stress the dependence of $\varepsilon(t)$ on θ:

$$\varepsilon(t,\theta) = H(z,\theta)v(t) \tag{B.5.3}$$

Because the polynomials of the ARMA model are monic by assumption, we have

$$H(z,\theta)|_{z^{-1}=0} = I \qquad \text{(for any } \theta) \tag{B.5.4}$$

This observation, along with the fact that $\varepsilon(t)$ is white and the "whitening filter" $H(z)$ is stable and causal (which follows from the fact that the complex-valued (equivalent) counterpart of (B.5.2), $e(t) = \frac{A(z)}{B(z)}y(t)$, is stable and causal), implies that (B.5.3) is a standard *prediction error* model, to which the CRB result of [Söderström and Stoica 1989] applies.

Let

$$\Delta(t) = \frac{\partial \varepsilon^T(t,\theta)}{\partial \theta} \tag{B.5.5}$$

($\varepsilon(t,\theta)$ depends on θ via $H(z,\theta)$ only; see (B.5.2)). Then, an asymptotically valid expression for the CRB block corresponding to the parameters in θ is given by

$$\boxed{P_{cr,\theta} = \left(E\left\{\Delta(t)\Lambda^{-1}\Delta^T(t)\right\}\right)^{-1}} \tag{B.5.6}$$

The calculation of the derivative matrix in (B.5.5) is straightforward. The evaluation of the statistical expectation in (B.5.6) can be reduced to ARMA cross-covariance calculations. Equation (B.5.6) does not require handling matrices of large dimensions (on the order of N), so its implementation is much simpler than that of the exact CRB formula.

For some recent results on the CRB for rational spectral analysis, see [Ninness 2003].

B.6 THE CRB FOR SPATIAL SPECTRA

Consider the model (6.2.21) for the output sequence $\{y(t)\}_{t=1}^{N}$ of an array that receives the signals emitted by n narrowband point sources:

$$\begin{aligned} y(t) &= As(t) + e(t) \\ A &= [a(\theta_1), \ldots, a(\theta_n)] \end{aligned} \tag{B.6.1}$$

The noise term, $e(t)$, in (B.6.1) is assumed to be circularly Gaussian distributed, with mean zero and the following covariances:

$$E\left\{e(t)e^*(\tau)\right\} = \sigma^2 I \delta_{t,\tau} \tag{B.6.2}$$

Regarding the signal vector, $s(t)$, in the equation (B.6.1), we can assume that either

Det: $\{s(t)\}$ is a deterministic, unknown sequence

or

Sto: $\{s(t)\}$ is a random sequence that is circularly Gaussian distributed with mean zero and covariances

$$E\left\{s(t)s^*(\tau)\right\} = P\delta_{t,\tau} \tag{B.6.3}$$

Hereafter, the acronyms Det and Sto are used to designate the case of deterministic or stochastic signals, respectively. Note that making one of these two assumptions on $\{s(t)\}$ is similar to assuming in the line-spectral analysis problem that the initial phases $\{\varphi_k\}$ are deterministic or random. (See Section B.4.) As we will see shortly, both the CRB analysis and the resulting CRB formulas depend heavily on which of the two assumptions we make on $\{s(t)\}$. The reader may already wonder which assumption should then be used in a given application. This is not a simple question, and we will be better prepared to answer it after deriving the corresponding CRB formulas.

In Chapter 6, we used the symbol θ to denote the DOA vector. To conform with the notation used in this appendix (and by a slight abuse of notation), we will here let θ denote the *entire* parameter vector.

As explained in Chapter 6, the use of array processing for spatial spectral analysis leads essentially to a parameter estimation problem. Under *the Det assumption* the parameter vector to be estimated is given by

$$\theta = \left[\theta_1, \ldots, \theta_n \,;\, \bar{s}^T(1), \ldots, \bar{s}^T(N)\,;\, \ldots;\, \tilde{s}^T(1), \ldots, \tilde{s}^T(N)\,;\, \sigma^2\right]^T \tag{B.6.4}$$

whereas under *the Sto assumption*

$$\theta = \left[\theta_1, \ldots, \theta_n \,;\, P_{11}, \bar{P}_{12}, \tilde{P}_{12}, \ldots, \bar{P}_{1n}, \tilde{P}_{1n}, P_{22}, \bar{P}_{23}, \tilde{P}_{23}, \ldots, P_{nn},\,;\, \sigma^2\right]^T \tag{B.6.5}$$

Hereafter, $\bar{s}(t)$ and $\tilde{s}(t)$ denote the real and imaginary parts of $s(t)$, and P_{ij} denotes the (i, j)th element of the matrix P. Furthermore, under both Det and Sto assumptions, the observed array output sample,

$$y(t) = \left[y^T(1), \ldots, y^T(N)\right]^T \tag{B.6.6}$$

is circularly Gaussian distributed with the following mean μ and covariance Γ:

Under Det:

$$\mu = \begin{bmatrix} As(1) \\ \vdots \\ As(N) \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} \sigma^2 I & & 0 \\ & \ddots & \\ 0 & & \sigma^2 I \end{bmatrix} \tag{B.6.7}$$

Under Sto:

$$\mu = 0, \qquad \Gamma = \begin{bmatrix} R & & 0 \\ & \ddots & \\ 0 & & R \end{bmatrix} \tag{B.6.8}$$

where R is given by (see ()6.4.3)

$$R = APA^* + \sigma^2 I \tag{B.6.9}$$

The differentiation of either (B.6.7) or (B.6.8) with respect to the elements of the parameter vector θ is straightforward. Use of the so-obtained derivatives of μ and Γ in the general CRB formula in (B.3.25) provides a simple means of computing CRB_{Det} and CRB_{Sto} for the entire parameter vector θ as defined in (B.6.4) or (B.6.5).

Computing the CRB as just described may be sufficient for many applications. However, sometimes we may need more than just that. For example, we may be interested in using the CRB for the design of array geometry or for getting insights into the various features of a specific spatial spectral analysis scenario. In such cases, we might want to have a closed-form (or analytical) expression for the CRB. More precisely, as the DOAs are usually the parameters of major interest, we often will want a closed-form expression for CRB(DOA) (i.e., the block of the CRB matrix that corresponds to the DOA parameters). Next, we consider the problem of obtaining such a closed-form CRB expression under both the Det and Sto assumptions just made.

First, consider the Det assumption. Let us write the corresponding μ vector in (B.6.7) as

$$\mu = Gs \tag{B.6.10}$$

where

$$G = \begin{bmatrix} A & & 0 \\ & \ddots & \\ 0 & & A \end{bmatrix}, \qquad s = \begin{bmatrix} s(1) \\ \vdots \\ s(N) \end{bmatrix} \tag{B.6.11}$$

Then, a straightforward calculation yields

$$\frac{\partial \mu}{\partial \bar{s}^T} = G, \qquad \frac{\partial \mu}{\partial \tilde{s}^T} = iG; \tag{B.6.12}$$

and

$$\frac{\partial \mu}{\partial \theta_k} = \begin{bmatrix} \frac{\partial A}{\partial \theta_k} s(1) \\ \vdots \\ \frac{\partial A}{\partial \theta_k} s(N) \end{bmatrix} = \begin{bmatrix} d_k s_k(1) \\ \vdots \\ d_k s_k(N) \end{bmatrix}, \qquad k = 1, \ldots, n \tag{B.6.13}$$

where $s_k(t)$ is the kth element of $s(t)$ and

$$d_k = \left. \frac{\partial a(\theta)}{\partial \theta} \right|_{\theta = \theta_k} \tag{B.6.14}$$

Using the notation

$$\Delta = \begin{bmatrix} d_1 s_1(1) & \cdots & d_n s_n(1) \\ \vdots & & \vdots \\ d_1 s_1(N) & \cdots & d_n s_n(N) \end{bmatrix}, \qquad (N \times n) \tag{B.6.15}$$

we can then write

$$\frac{d\mu}{d\theta^T} = [\Delta, G, iG, 0] \tag{B.6.16}$$

which gives the following expression for the second term in the general CRB formula in (B.3.25):

$$2 \operatorname{Re} \left\{ \frac{d\mu^*}{d\theta} \Gamma^{-1} \frac{d\mu}{d\theta^T} \right\} = \begin{bmatrix} J & 0 \\ 0 & 0 \end{bmatrix} \tag{B.6.17}$$

In this equation

$$J \triangleq \frac{2}{\sigma^2} \operatorname{Re} \left\{ \begin{bmatrix} \Delta^* \\ G^* \\ -iG^* \end{bmatrix} [\Delta \ \ G \ \ iG] \right\} \tag{B.6.18}$$

Furthermore, Γ depends only on σ^2, and

$$\frac{d\Gamma}{d\sigma^2} = \begin{bmatrix} I & & 0 \\ & \ddots & \\ 0 & & I \end{bmatrix}$$

so we can easily verify that the matrix corresponding to the first term in the general CRB formula, (B.3.25), is given by

$$\operatorname{tr}\left[\Gamma^{-1}\Gamma_i'\Gamma^{-1}\Gamma_j'\right] = \begin{bmatrix} 0 & 0 \\ 0 & \frac{mN}{\sigma^4} \end{bmatrix}, \qquad i, j = 1, 2, \ldots \tag{B.6.19}$$

Combining (B.6.17) and (B.6.19) yields the following CRB formula for the parameter vector θ in (B.6.4), under the Det assumption:

$$\text{CRB}_{\text{Det}} = \begin{bmatrix} J^{-1} & 0 \\ 0 & \frac{\sigma^4}{mN} \end{bmatrix} \tag{B.6.20}$$

Hence, to obtain the CRB for the DOA subvector of θ, we need to extract the corresponding block of J^{-1}. One convenient way of doing this is by suitably block-diagonalizing the matrix J. To this end, let us introduce the matrix

$$B = (G^*G)^{-1}G^*\Delta \tag{B.6.21}$$

Note that the inverse in (B.6.21) exists, because A^*A is nonsingular by assumption. Also, let

$$F = \begin{bmatrix} I & 0 & 0 \\ -\bar{B} & I & 0 \\ -\tilde{B} & 0 & I \end{bmatrix} \tag{B.6.22}$$

where $\bar{B} = \text{Re}\{B\}$ and $\tilde{B} = \text{Im}\{B\}$. It can be verified that

$$\begin{bmatrix} \Delta & G & iG \end{bmatrix} F = \begin{bmatrix} (\Delta - GB) & G & iG \end{bmatrix} = \begin{bmatrix} \Pi_G^{\perp}\Delta & G & iG \end{bmatrix} \tag{B.6.23}$$

where

$$\Pi_G^{\perp} = I - G(G^*G)^{-1}G^*$$

is the orthogonal projector onto the null space of G^* (see Result R17 in Appendix A); in particular, observe that $G^*\Pi_G^{\perp} = 0$. It follows from (B.6.18) and (B.6.23) that

$$\begin{aligned} F^T J F &= \frac{2}{\sigma^2}\,\text{Re}\left\{ F^* \begin{bmatrix} \Delta^* \\ G^* \\ -iG^* \end{bmatrix} \begin{bmatrix} \Delta & G & iG \end{bmatrix} F \right\} \\ &= \frac{2}{\sigma^2}\,\text{Re}\left\{ \begin{bmatrix} \Delta^*\Pi_G^{\perp} \\ G^* \\ -iG^* \end{bmatrix} \begin{bmatrix} \Pi_G^{\perp}\Delta & G & iG \end{bmatrix} \right\} \\ &= \frac{2}{\sigma^2}\,\text{Re}\left\{ \begin{bmatrix} \Delta^*\Pi_G^{\perp}\Delta & 0 & 0 \\ 0 & G^*G & iG^*G \\ 0 & -iG^*G & G^*G \end{bmatrix} \right\} \end{aligned} \tag{B.6.24}$$

and hence, that the CRB matrix for the DOAs and the signal sequence is given by

$$\begin{aligned}
J^{-1} &= F\left(F^T J F\right)^{-1} F^T \\
&= \frac{\sigma^2}{2}\begin{bmatrix} I & 0 & 0 \\ -\bar{B} & I & 0 \\ -\tilde{B} & 0 & I \end{bmatrix}\begin{bmatrix} \left[\operatorname{Re}(\Delta^* \Pi_G^{\perp} \Delta)\right]^{-1} & 0 & 0 \\ 0 & x & x \\ 0 & x & x \end{bmatrix}\begin{bmatrix} I & -\bar{B}^T & -\tilde{B}^T \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \\
&= \begin{bmatrix} \frac{\sigma^2}{2}\left[\operatorname{Re}(\Delta^* \Pi_G^{\perp} \Delta)\right]^{-1} & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \qquad \text{(B.6.25)}
\end{aligned}$$

where we used the symbol x to denote a block of no interest in the derivation. From (B.6.4) and (B.6.25), we can immediately see that the CRB matrix for the DOAs is given by

$$\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA}) = \frac{\sigma^2}{2}\left[\operatorname{Re}(\Delta^* \Pi_G^{\perp} \Delta)\right]^{-1} \qquad \text{(B.6.26)}$$

It is possible to rewrite (B.6.26) in a more convenient form. To do so, we note that

$$\Pi_G^{\perp} = \begin{bmatrix} I & & 0 \\ & \ddots & \\ 0 & & I \end{bmatrix} - \begin{bmatrix} \Pi_A & & 0 \\ & \ddots & \\ 0 & & \Pi_A \end{bmatrix} = \begin{bmatrix} \Pi_A^{\perp} & & 0 \\ & \ddots & \\ 0 & & \Pi_A^{\perp} \end{bmatrix} \qquad \text{(B.6.27)}$$

and, hence, that

$$\begin{aligned}
\left[\Delta^* \Pi_G^{\perp} \Delta\right]_{kp} &= \sum_{t=1}^{N} d_k^* s_k^*(t) \Pi_A^{\perp} d_p s_p(t) \\
&= N\left[d_k^* \Pi_A^{\perp} d_p\right]\left[\frac{1}{N}\sum_{t=1}^{N} s_p(t) s_k^*(t)\right] \\
&= N\left[D^* \Pi_A^{\perp} D\right]_{kp}\left[\hat{P}^T\right]_{kp} \qquad \text{(B.6.28)}
\end{aligned}$$

where

$$D = \left[d_1 \ \dots \ d_n\right] \qquad \text{(B.6.29)}$$

$$\hat{P} = \frac{1}{N}\sum_{t=1}^{N} s(t) s^*(t) \qquad \text{(B.6.30)}$$

It follows from (B.6.28) that

$$\Delta^* \Pi_G^{\perp} \Delta = N \ \left(D^* \Pi_A^{\perp} D\right) \odot \hat{P}^T \qquad \text{(B.6.31)}$$

where $\odot$ denotes the Hadamard (or elementwise) matrix product, defined in Result R19 in Appendix A. Inserting (B.6.31) in (B.6.26) yields the following analytical expression for *the CRB matrix associated with the DOA vector under the Det assumption:*

$$\boxed{\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA}) = \frac{\sigma^2}{2N}\left\{\mathrm{Re}\left[\left(D^*\Pi_A^{\perp}D\right)\odot \hat{P}^T\right]\right\}^{-1}} \tag{B.6.32}$$

We refer the reader to [Stoica and Nehorai 1989a] for more details about (B.6.32) and its possible uses in array processing. The presented derivation of (B.6.32) has been adapted from [Stoica and Larsson 2001]. Note that (B.6.32) can be applied directly to the temporal line-spectral model in Section B.4 (see equations (B.4.4) and (B.4.5)) to obtain an analytical CRB formula for the sinusoidal frequencies.

The derivation of an analytical expression for *the CRB matrix associated with the DOAs under the Sto assumption* is more intricate, and we give only the final formula here (see [Stoica, Larsson, and Gershman 2001] and its references for a derivation):

$$\boxed{\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA}) = \frac{\sigma^2}{2N}\left\{\mathrm{Re}\left[\left(D^*\Pi_A^{\perp}D\right)\odot \left(PA^*R^{-1}AP\right)^T\right]\right\}^{-1}} \tag{B.6.33}$$

At this point, we should emphasize the fact that the two CRBs, $\mathrm{CRB}_{\mathrm{Det}}$ and $\mathrm{CRB}_{\mathrm{Sto}}$, correspond to two different models of the data vector y (see (B.6.7) and (B.6.8)); hence, they are *not* directly comparable. On the other hand, the CRBs for the DOA parameters can be compared with one another. To make this comparison possible, let us introduce the assumption that the sample covariance matrix $\hat{P}$ in (B.6.30) converges to the P matrix in (B.6.3), as $N \to \infty$. Let $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$ denote the CRB matrix in (B.6.32) with $\hat{P}$ replaced by P. Then, the following interesting order relation holds true:

$$\boxed{\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA}) \geq \overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})} \tag{B.6.34}$$

To prove (B.6.34), we need to show (see (B.6.32) and (B.6.33)) that

$$\left\{\mathrm{Re}\left[\left(D^*\Pi_A^{\perp}D\right)\odot \left(PA^*R^{-1}AP\right)^T\right]\right\}^{-1} \geq \left\{\mathrm{Re}\left[\left(D^*\Pi_A^{\perp}D\right)\odot P^T\right]\right\}^{-1}$$

or, equivalently, that

$$\mathrm{Re}\left[\left(D^*\Pi_A^{\perp}D\right)\odot \left(P - PA^*R^{-1}AP\right)^T\right] \geq 0 \tag{B.6.35}$$

The real part of a positive semidefinite matrix is positive semidefinite itself:

$$H \geq 0 \implies \mathrm{Re}[H] \geq 0 \tag{B.6.36}$$

(Indeed, for any real-valued vector h we have: $h^* \mathrm{Re}[H]h = \mathrm{Re}[h^* H h] \geq 0$ for $H \geq 0$.) Combining this observation with Result R19 in Appendix A shows that, to prove (B.6.35), it is sufficient to verify that

$$P \geq PA^* R^{-1} AP \tag{B.6.37}$$

or, equivalently,

$$I \geq P^{1/2} A^* R^{-1} A P^{1/2} \tag{B.6.38}$$

where $P^{1/2}$ denotes the Hermitian square root of P; see Definition D12 in Appendix A. Let

$$Z = AP^{1/2}$$

Then (B.6.38) can be rewritten as

$$I - Z^* \left(ZZ^* + \sigma^2 I\right)^{-1} Z \geq 0 \tag{B.6.39}$$

To prove (B.6.39), we use the fact that the following matrix is evidently positive semidefinite:

$$\begin{bmatrix} I & Z^* \\ Z & ZZ^* + \sigma^2 I \end{bmatrix} = \begin{bmatrix} I \\ Z \end{bmatrix} \begin{bmatrix} I & Z^* \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \sigma^2 I \end{bmatrix} \geq 0 \tag{B.6.40}$$

and therefore

$$\begin{aligned} &\begin{bmatrix} I & -Z^* \left(ZZ^* + \sigma^2 I\right)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} I & Z^* \\ Z & ZZ^* + \sigma^2 I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\left(ZZ^* + \sigma^2 I\right)^{-1} Z & I \end{bmatrix} \\ &= \begin{bmatrix} I - Z^* \left(ZZ^* + \sigma^2 I\right)^{-1} Z & 0 \\ 0 & ZZ^* + \sigma^2 I \end{bmatrix} \geq 0 \end{aligned} \tag{B.6.41}$$

The inequality in (B.6.39) is a simple consequence of (B.6.41), and so the proof of (B.6.34) is concluded.

To understand (B.6.34) at an *intuitive level*, we note that the ML method for DOA estimation under the Sto assumption, $\mathrm{ML}_{\mathrm{Sto}}$, can be shown to achieve $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$ (for sufficiently large values of N). (See, e.g., [Stoica and Nehorai 1990] and [Ottersten, Viberg, Stoica, and Nehorai 1993].) This result should in fact be no surprise, because the general ML method of parameter estimation is known to be asymptotically statistically efficient (i.e., it achieves the CRB as $N \to \infty$) under some regularity conditions that are satisfied in the Sto assumption case. Specifically, the regularity conditions require that the number of unknown parameters not increase as N increases, as is indeed true for the Sto model (see (B.6.5)). Let $\mathrm{CML}_{\mathrm{Sto}}(\mathrm{DOA})$ denote the asymptotic covariance matrix of the $\mathrm{ML}_{\mathrm{Sto}}$ estimate of the DOA parameter vector. According to the preceding discussion, we have that

$$\mathrm{CML}_{\mathrm{Sto}}(\mathrm{DOA}) = \mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA}) \tag{B.6.42}$$

At the same time, under the Det assumption, the $\mathrm{ML}_{\mathrm{Sto}}$ can be viewed as *some* method for DOA estimation, and hence its asymptotic covariance matrix must satisfy the CRB inequality (corresponding to the Det assumption):

$$\mathrm{CML}_{\mathrm{Sto}}(\mathrm{DOA}) \geq \overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA}) \tag{B.6.43}$$

(Note that the asymptotic covariance matrix of $\mathrm{ML}_{\mathrm{Sto}}$ can be shown to be the same under either the Sto or Det assumption.) This equation, along with (B.6.42), provides a heuristic motivation for the relationship between $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$ and $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$ in (B.6.34). Note that the inequality in (B.6.34) is, in general, *strict*, but the relative difference between $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$ and $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$ is usually fairly small. (See, e.g., [OTTERSTEN, VIBERG, STOICA, AND NEHORAI 1993].)

A remark similar to the one in the previous paragraph can be made on the ML method for DOA estimation under the Det assumption, which we abbreviate as $\mathrm{ML}_{\mathrm{Det}}$. Note that $\mathrm{ML}_{\mathrm{Det}}$ can be readily seen to coincide with the *NLS method* discussed in Section 6.4.1. Under the Sto assumption, $\mathrm{ML}_{\mathrm{Det}}$ (i.e., the NLS method) can be viewed as just *some* method for DOA estimation. Hence, its (asymptotic) covariance matrix must be bounded below by the CRB corresponding to the Sto assumption:

$$\mathrm{CML}_{\mathrm{Det}}(\mathrm{DOA}) \geq \mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA}) \tag{B.6.44}$$

Like $\mathrm{ML}_{\mathrm{Sto}}$, the asymptotic covariance matrix of $\mathrm{ML}_{\mathrm{Det}}$ can also be shown to be the same under either the Sto or Det assumption. Hence, we can infer from (B.6.34) and (B.6.44) that *ML_{Det} does not attain $\overline{CRB}_{Det}(DOA)$*, as is indeed the case (as is shown in, e.g., [STOICA AND NEHORAI 1989A]). To understand why this happens, note that the Det model contains $(2N+1)n+1$ real-valued parameters (see (B.6.4)), which must be estimated from $2mN$ data samples. Hence, for large N, the ratio between the number of unknown parameters and the available data samples approaches a constant (equal to n/m), which violates one of the aforementioned regularity conditions for the statistical efficiency of the ML method.

Remark: $\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA})$ depends on the signal sequence $\{s(t)\}_{t=1}^{N}$. However, neither $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$ nor the asymptotic covariance matrix of $\mathrm{ML}_{\mathrm{Sto}}$, of $\mathrm{ML}_{\mathrm{Det}}$, or, in fact, of many other DOA estimation methods depends on this sequence. We will use the symbol C to denote the (asymptotic) covariance matrix of such a DOA estimation method for which C is independent of the signal sequence.

From $\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA})$ we can obtain a matrix, different from $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$, which is independent of the signal sequence, in the following manner:

$$\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA}) = \tilde{E}\left\{\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA})\right\} \tag{B.6.45}$$

Here $\tilde{E}$ is an averaging operator and $\mathrm{ACRB}_{\mathrm{Det}}$ stands for Averaged $\mathrm{CRB}_{\mathrm{Det}}$. For example, $\tilde{E}\{\cdot\}$ in (B.6.45) can be a simple arithmetic averaging of $\mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA})$ over a set of signal sequences. Using the fact that $\tilde{E}\{C\} = C$ (because C does not depend on the sequence $\{s(t)\}_{t=1}^{N}$), we can apply the operator $\tilde{E}\{\cdot\}$ to both sides of the CRB inequality

$$C \geq \mathrm{CRB}_{\mathrm{Det}}(\mathrm{DOA}) \tag{B.6.46}$$

to obtain

$$C \geq \mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA}) \tag{B.6.47}$$

(Note that the inequality in (B.6.47) and, hence, that in (B.6.47), hold at least for sufficiently large values of N.) It follows from (B.6.47) that $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$ can also be used as a lower bound on the DOA estimation error covariance. Furthermore, it can be shown that $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$ is *tighter* than $\overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})$:

$$\boxed{\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA}) \geq \overline{\mathrm{CRB}}_{\mathrm{Det}}(\mathrm{DOA})} \tag{B.6.48}$$

To prove (B.6.48), we introduce the matrix

$$X = \frac{2N}{\sigma^2} \operatorname{Re}\left[(D^* \Pi_A^{\perp} D) \odot \hat{P}^T\right] \tag{B.6.49}$$

Using this notation, along with the fact that $\tilde{E}\{\hat{P}\} = P$ (which holds under mild conditions), we can rewrite (B.6.48) as follows:

$$\boxed{\tilde{E}\left\{X^{-1}\right\} \geq \left[\tilde{E}\left\{X\right\}\right]^{-1}} \tag{B.6.50}$$

To prove (B.6.50), we note that the matrix

$$\tilde{E}\left\{\begin{bmatrix} X^{-1} & I \\ I & X \end{bmatrix}\right\} = \tilde{E}\left\{\begin{bmatrix} X^{-1/2} \\ X^{1/2} \end{bmatrix}\begin{bmatrix} X^{-1/2} & X^{1/2} \end{bmatrix}\right\}$$

(where $X^{1/2}$ and $X^{-1/2}$ denote the Hermitian square roots of X and X^{-1}, respectively) is clearly positive semidefinite, and therefore so must be the following matrix:

$$\begin{aligned} &\begin{bmatrix} I & -\left[\tilde{E}\left\{X\right\}\right]^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} \tilde{E}\left\{X^{-1}\right\} & I \\ I & \tilde{E}\left\{X\right\} \end{bmatrix}\begin{bmatrix} I & 0 \\ -\left[\tilde{E}\left\{X\right\}\right]^{-1} & I \end{bmatrix} \\ &\quad = \begin{bmatrix} \tilde{E}\left\{X^{-1}\right\} - \left[\tilde{E}\left\{X\right\}\right]^{-1} & 0 \\ 0 & \tilde{E}\left\{X\right\} \end{bmatrix} \geq 0 \end{aligned} \tag{B.6.51}$$

The matrix inequality in (B.6.50), which is somewhat similar to the scalar Jensen inequality (see, e.g., Complement 4.9.5) readily follows from (B.6.51).

The inequality (B.6.48) looks appealing. On the other hand, $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$ should be *less tight* than $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$, in view of the results in (B.6.42) and (B.6.47). Also, $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$ has a simpler analytical form. Hence, we may have little reason to use $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$ in lieu of $\mathrm{CRB}_{\mathrm{Sto}}(\mathrm{DOA})$. Despite these drawbacks of $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$, we have included this discussion for the potential usefulness of the inequality in (B.6.50) and of the basic idea behind the introduction of $\mathrm{ACRB}_{\mathrm{Det}}(\mathrm{DOA})$. ■

In the remainder of this section, we rely on the previous results to compare the Det and Sto model assumptions, to discuss the consequences of making these assumptions, and to draw some conclusions.

First, consider the array output model in equation (B.6.1). To derive the ML estimates of the unknown parameters in (B.6.1), we must make some assumptions on the signal sequence $\{s(t)\}$. The ML_{Sto} method for DOA estimation (derived under the Sto assumption) turns out to be more accurate than the ML_{Det} method (obtained under the Det assumption), under quite general conditions on $\{s(t)\}$. However, the ML_{Sto} method is somewhat more complicated computationally than the ML_{Det} method; see, e.g., [Ottersten, Viberg, Stoica, and Nehorai 1993].

The previous discussion implies that the question about which assumption should be used (because "it is more likely to be true") is in fact irrelevant in this case. Indeed, we should see the two assumptions only as instruments for deriving the two corresponding ML methods. Once we have completed the derivations, the assumption issue is no longer important, and we can simply *choose the ML method that we prefer, regardless of the nature of* $\{s(t)\}$. The choice should be based on the facts that (a) ML_{Det} is computationally simpler than ML_{Sto}, and (b) ML_{Sto} is statistically more accurate than ML_{Det} under quite general conditions on $\{s(t)\}$.

Second, regarding the two CRB matrices that correspond to the Det and Sto assumptions, respectively, we can argue as follows: Under the Sto assumption, $\text{CRB}_{\text{Sto}}(\text{DOA})$ is *the* Cramér–Rao bound and, hence, the lower bound to use. Under the Det assumption, $\text{CRB}_{\text{Sto}}(\text{DOA})$ is no longer the true CRB, but it is still a tight lower bound on the asymptotic covariance matrix of any known DOA estimation method. $\overline{\text{CRB}}_{\text{Det}}(\text{DOA})$ is also a lower bound, but it is not tight. Hence, *$\text{CRB}_{\text{Sto}}(\text{DOA})$ should be the normal choice for a lower bound, regardless of the assumption (Det or Sto) that the signal sequence is likely to satisfy*. Note that, under the Det assumption, ML_{Sto} can be seen as *some* DOA estimation method. Therefore, in principle, a better DOA estimation method than ML_{Sto} could exist (where by "better" we mean that the covariance matrix of such an estimation method would be smaller than $\text{CRB}_{\text{Sto}}(\text{DOA})$). However, no such DOA estimation method appears to be available, in spite of a significant literature on the so-called problem of "estimation in the presence of many nuisance parameters," of which the DOA estimation problem under the Det assumption is a special case.

Appendix C

Model Order Selection Tools

C.1 INTRODUCTION

The parametric methods of spectral analysis (discussed in Chapters 3, 4, and 6) require not only the estimation of a vector of real-valued parameters but also the selection of one or several *integer-valued* parameters that are equally important for the specification of the data model. Specifically, these integer-valued parameters of the model are the ARMA model orders (in Chapter 3), the number of sinusoidal components (in Chapter 4), and the number of source signals impinging on the array (in Chapter 6). In each of these cases, the integer-valued parameters determine the dimension of the real-valued parameter vector of the data model. In what follows, we will use the following symbols:

$$y = \text{the vector of available data (of size } N)$$

$$\theta = \text{the (real-valued) parameter vector}$$

$$n = \text{the dimension of } \theta$$

For short, we will refer to n as the *model order*, even though sometimes n is not really an order. (See, for example, the preceding examples.) We assume that both y and θ are real valued:

$$y \in \mathbf{R}^N, \qquad \theta \in \mathbf{R}^n$$

Whenever we need to emphasize that the number of elements in θ is n, we will use the notation θ^n. A method that estimates n from the data vector y will be called an *order-selection rule*. Note that the need for estimating a model order is typical of the parametric approaches to spectral analysis. The nonparametric methods of spectral analysis do not have such a requirement.

The discussion in the text on the parametric spectral methods has focused on estimating the model-parameter vector θ for a specific order n. In this general appendix (based on [Stoica and Selén 2004b]) we explain how to estimate n as well. The literature on order selection is as considerable as that on (real-valued) parameter estimation (see, e.g., [Choi 1992; Söderström and Stoica 1989; McQuarrie and Tsai 1998; Linhart and Zucchini 1986; Burnham and Anderson 2002; Sakamoto, Ishiguro, and Kitagawa 1986; Stoica, Eykhoff, Jannsen, and Söderström 1986] and the many references therein). However, many order selection rules are tied to specific parameter estimation methods; hence, their applicability is rather limited. Here we will concentrate on order-selection rules that are associated with the maximum likelihood method (MLM) of parameter estimation. As explained briefly in Appendix B (and in what follows here), the MLM is likely the most commonly used parameter estimation method. Consequently, the order estimation rules that can be used with the MLM are of quite a general interest. In the next section, we review briefly the ML method of parameter estimation and some of its main properties.

C.2 MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

Let

> $p(y, \theta) =$ the probability density function (pdf) of the data vector y, which depends on the parameter vector θ; also called *the likelihood function.*

The ML estimate of θ, which we denote by $\hat{\theta}$, is given by the maximizer of $p(y, \theta)$ (see, for example, [Anderson 1971; Brockwell and Davis 1991; Hannan and Deistler 1988; Papoulis 1977; Porat 1994; Priestley 1981; Scharf 1991; Therrien 1992; Söderström and Stoica 1989] and Appendix B). Alternatively, because $\ln(\cdot)$ is a monotonically increasing function,

$$\boxed{\hat{\theta} = \arg\max_{\theta} \ \ln p(y, \theta)} \tag{C.2.1}$$

Under the Gaussian data assumption, the MLM typically reduces to the nonlinear least-squares (NLS) method of parameter estimation (particular forms of which are discussed briefly in Chapter 3 and in more detail in Chapters 4 and 6). To illustrate this fact, let us assume that the observation vector y can be written as

$$y = \mu(\gamma) + e \tag{C.2.2}$$

where e is a (real-valued) Gaussian white-noise vector with mean zero and covariance matrix given by $E\left\{ee^T\right\} = \sigma^2 I$, γ is an unknown parameter vector, and $\mu(\gamma)$ is a deterministic function of γ. It follows readily from (C.2.2) that

$$p(y,\theta) = \frac{1}{(2\pi)^{N/2}(\sigma^2)^{N/2}} e^{-\frac{\|y-\mu(\gamma)\|^2}{2\sigma^2}} \tag{C.2.3}$$

where

$$\theta = \begin{bmatrix} \gamma \\ \sigma^2 \end{bmatrix} \tag{C.2.4}$$

Remark: Note that, in this appendix, we use the symbol θ for the whole parameter vector, unlike in some previous discussions, where we used θ to denote the signal parameter vector (which is denoted by γ here). ■

We deduce from (C.2.3) that

$$-2\ln p(y,\theta) = N\ln(2\pi) + N\ln\sigma^2 + \frac{\|y-\mu(\gamma)\|^2}{\sigma^2} \tag{C.2.5}$$

A simple calculation based on (C.2.5) shows that the ML estimates of γ and σ^2 are given by

$$\hat{\gamma} = \arg\min_{\gamma} \|y-\mu(\gamma)\|^2 \tag{C.2.6}$$

$$\hat{\sigma}^2 = \frac{1}{N}\|y-\mu(\hat{\gamma})\|^2 \tag{C.2.7}$$

The corresponding value of the likelihood function is given by

$$\boxed{-2\ln p(y,\hat{\theta}) = \text{constant} + N\ln\hat{\sigma}^2} \tag{C.2.8}$$

As can be seen from (C.2.6), in the present case the MLM indeed reduces to the NLS. In particular, note that the NLS method for sinusoidal parameter estimation discussed in Chapter 4 is precisely of the form of (C.2.6). If we let N_s denote the number of observed complex-valued samples of the noisy sinusoidal signal and n_c denote the number of sinusoidal components present in the signal, then

$$N = 2N_s \tag{C.2.9}$$

$$n = 3n_c + 1 \tag{C.2.10}$$

We will use the sinusoidal signal model of Chapter 4 as a vehicle for illustrating how the various general order-selection rules presented in what follows should be used in a specific situation. These rules can also be used with the parametric spectral analysis methods of Chapters 3 and 6. The task of deriving *explicit forms* of these order selection rules for the aforementioned methods is left as an interesting exercise to the reader (see, for example, [McQuarrie and Tsai 1998; Brockwell and Davis 1991; Porat 1994]).

Next, we note that, under regularity conditions, the pdf of the ML estimate $\hat{\theta}$ converges, as $N \to \infty$, to a Gaussian pdf with mean θ and covariance matrix equal to the Cramér–Rao bound (CRB) matrix (see Section B.2 for a discussion about the CRB). Consequently, asymptotically in N, the pdf of $\hat{\theta}$ is given by

$$p(\hat{\theta}) = \frac{1}{(2\pi)^{n/2}|J^{-1}|^{1/2}} e^{-\frac{1}{2}(\hat{\theta}-\theta)^T J(\hat{\theta}-\theta)} \tag{C.2.11}$$

where (see (B.2.10))

$$J = -E\left\{\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\, \partial\theta^T}\right\} \tag{C.2.12}$$

Remark: To simplify the notation, we use the symbol θ for both the true parameter vector and the parameter vector viewed as an unknown variable (as we also did in Appendix B). The exact meaning of θ should be clear from the context. ■

The "regularity conditions" referred to previously require that n not be a function of N and, hence, that the ratio between the number of unknown parameters and the number of observations tends to zero as $N \to \infty$. This is true for the parametric spectral analysis problems discussed in Chapters 3 and 4. However, the previous condition does not hold for the parametric spectral analysis problem addressed in Chapter 6. Indeed, in the latter case, the number of parameters to be estimated from the data is proportional to N, because the signal sequence is completely unknown. To overcome this difficulty, we can assume that the signal vector is temporally white and Gaussian distributed, which leads to a ML problem that satisfies the previously stated regularity condition. (We refer the interested reader to [Ottersten, Viberg, Stoica, and Nehorai 1993; Stoica and Nehorai 1990; Van Trees 2002] for details on this ML approach to the spatial spectral analysis problem of Chapter 6.)

To close this section, we note that, under mild conditions,

$$\left[-\frac{1}{N}\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\, \partial\theta^T} - \frac{1}{N}J\right] \to 0 \quad \text{as } N \to \infty \tag{C.2.13}$$

To motivate (C.2.13) for the fairly general data model in (C.2.2), we can argue as follows: Let us rewrite the negative log-likelihood function associated with (C.2.2) (see (C.2.5)) as

$$-\ln p(y,\theta) = \text{constant} + \frac{N}{2}\ln(\sigma^2) + \frac{1}{2\sigma^2}\sum_{t=1}^{N}\left[y_t - \mu_t(\gamma)\right]^2 \tag{C.2.14}$$

where the subindex t denotes the t-th component. From (C.2.14), we obtain, by a simple calculation,

$$-\frac{\partial \ln p(y,\theta)}{\partial\theta} = \begin{bmatrix} -\dfrac{1}{\sigma^2}\displaystyle\sum_{t=1}^{N}\left[y_t - \mu_t(\gamma)\right]\mu_t'(\gamma) \\ \dfrac{N}{2\sigma^2} - \dfrac{1}{2\sigma^4}\displaystyle\sum_{t=1}^{N}\left[y_t - \mu_t(\gamma)\right]^2 \end{bmatrix} \tag{C.2.15}$$

where

$$\mu_t'(\gamma) = \frac{\partial \mu_t(\gamma)}{\partial \gamma} \tag{C.2.16}$$

Differentiating (C.2.15) once again gives

$$-\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\ \partial\theta^T}$$
$$= \begin{bmatrix} -\frac{1}{\sigma^2}\sum_{t=1}^{N} e_t\mu_t''(\gamma) + \frac{1}{\sigma^2}\sum_{t=1}^{N}\mu_t'(\gamma)\mu_t'^T(\gamma) & \frac{1}{\sigma^4}\sum_{t=1}^{N} e_t\mu_t'(\gamma) \\ \frac{1}{\sigma^4}\sum_{t=1}^{N} e_t\mu_t'(\gamma) & -\frac{N}{2\sigma^4} + \frac{1}{\sigma^6}\sum_{t=1}^{N} e_t^2 \end{bmatrix} \tag{C.2.17}$$

where $e_t = y_t - \mu_t(\gamma)$ and

$$\mu_t''(\gamma) = \frac{\partial^2 \mu_t(\gamma)}{\partial\gamma\ \partial\gamma^T} \tag{C.2.18}$$

Taking the expectation of (C.2.17) and dividing by N, we get

$$\frac{1}{N}J = \begin{bmatrix} \frac{1}{\sigma^2}\left(\frac{1}{N}\sum_{t=1}^{N}\mu_t'(\gamma)\mu_t'^T(\gamma)\right) & 0 \\ 0 & \frac{1}{2\sigma^4} \end{bmatrix} \tag{C.2.19}$$

We assume that $\mu(\gamma)$ is such that the previous matrix has a finite limit as $N \to \infty$. Under this assumption and the previously-made assumption on e, we can also show from (C.2.17) that

$$-\frac{1}{N}\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\ \partial\theta^T}$$

converges (as $N \to \infty$) to the right side of (C.2.19), which concludes the motivation of (C.2.13). Letting

$$\hat{J} = -\left.\frac{\partial^2 \ln p(y,\theta)}{\partial\theta\ \partial\theta^T}\right|_{\theta=\hat{\theta}} \tag{C.2.20}$$

we deduce from (C.2.13), (C.2.19), and the consistency of $\hat{\theta}$ that, for sufficiently large values of N,

$$\frac{1}{N}\hat{J} \simeq \frac{1}{N}J = \mathcal{O}(1) \tag{C.2.21}$$

Hereafter, $\simeq$ denotes an asymptotic (approximate) equality, in which the higher order terms have been neglected, and $\mathcal{O}(1)$ denotes a term that tends to a constant as $N \to \infty$.

Interestingly enough, the assumption that the right side of (C.2.19) has a finite limit, as $N \to \infty$, holds for many problems, but *not* for the sinusoidal parameter estimation problem of Chapter 4. In the latter case, (C.2.21) needs to be modified to (see, e.g., Appendix B)

$$K_N \hat{J} K_N \simeq K_N J K_N = \mathcal{O}(1) \tag{C.2.22}$$

where

$$K_N = \begin{bmatrix} \dfrac{1}{N_s^{3/2}} I_{n_c} & 0 \\ 0 & \dfrac{1}{N_s^{1/2}} I_{2n_c+1} \end{bmatrix} \tag{C.2.23}$$

and where I_k denotes the $k \times k$ identity matrix; to write (C.2.23), we assumed that the upper left $n_c \times n_c$ block of J corresponds to the sinusoidal frequencies, but this fact is not really important for the analysis in this appendix, as we will see below.

C.3 USEFUL MATHEMATICAL PRELIMINARIES AND OUTLOOK

In this section, we discuss a number of mathematical tools that will be used in the next sections to derive several important order-selection rules. We will keep the discussion at an informal level to make the material as accessible as possible. In Section C.3.1, we will formulate the model order selection as a hypothesis-testing problem, with the main goal of showing that the maximum *a posteriori* (MAP) approach leads to the optimal order-selection rule (in a sense specified there). In Section C.3.2, we discuss the Kullback–Leibler information criterion, which lies at the basis of another approach that can be used to derive model order selection rules.

C.3.1 Maximum *A Posteriori* (MAP) Selection Rule

Let H_n denote the hypothesis that the model order is n, and let $\bar{n}$ denote a known upper bound on n:

$$n \in [1, \bar{n}] \tag{C.3.1}$$

We assume that the hypotheses $\{H_n\}_{n=1}^{\bar{n}}$ are *mutually exclusive* (i.e., only one of them can hold true at a time). As an example, for a real-valued AR signal with coefficients $\{a_k\}$, we can define H_n as follows:

$$H_n: \; a_n \neq 0 \text{ and } a_{n+1} = \cdots = a_{\bar{n}} = 0 \tag{C.3.2}$$

For a sinusoidal signal we can proceed similarly, after observing that, for such a signal, the number of components n_c is related to n as in (C.2.10), *viz.*,

$$n = 3n_c + 1 \tag{C.3.3}$$

Hence, for a sinusoidal signal with amplitudes $\{\alpha_k\}$, we can consider the following hypotheses:

$$H_{n_c}: \ \alpha_k \neq 0 \text{ for } k = 1, \ldots, n_c, \ \text{ and } \alpha_k = 0 \text{ for } k = n_c + 1, \ldots, \bar{n}_c \tag{C.3.4}$$

for $n_c \in [1, \bar{n}_c]$ (with the corresponding "model order" n being given by (C.3.3)).

Remark: The hypotheses $\{H_n\}$ can be *either nested or non-nested.* We say that H_1 and H_2 are nested whenever the model corresponding to H_1 can be obtained as a special case of that associated with H_2. To give an example, the following hypotheses:

H_1 : the signal is a first-order AR process

H_2 : the signal is a second-order AR process

are nested, whereas the H_1 and

H_3 : the signal consists of one sinusoid in noise

are nonnested. ■

Let

$$p_n(y|H_n) = \text{the pdf of } y \text{ under } H_n \tag{C.3.5}$$

Whenever we want to emphasize the possible dependence of the pdf in (C.3.5) on the parameter vector of the model corresponding to H_n, we write

$$p_n(y, \theta^n) \triangleq p_n(y|H_n) \tag{C.3.6}$$

Assuming that (C.3.5) is available, along with the *a priori* probability of H_n, $p_n(H_n)$, we can write the conditional probability of H_n, given y, as

$$p_n(H_n|y) = \frac{p_n(y|H_n)p_n(H_n)}{p(y)} \tag{C.3.7}$$

The maximum *a posteriori* probability (MAP) rule selects the order n (or the hypothesis H_n) that maximizes (C.3.7). The denominator in (C.3.7) does not depend on n, so the *MAP rule* is given by

$$\boxed{\max_{n \in [1, \bar{n}]} p_n(y|H_n)p_n(H_n)} \tag{C.3.8}$$

Most typically, the hypotheses $\{H_n\}$ are *a priori equiprobable*—that is,

$$p_n(H_n) = \frac{1}{\bar{n}}, \quad n = 1, \ldots, \bar{n} \tag{C.3.9}$$

In such a case the MAP rule reduces to

$$\boxed{\max_{n\in[1,\bar{n}]} p_n(y|H_n)} \tag{C.3.10}$$

Next, we define the *average (or total) probability of correct detection* as

$$P_{cd} = \Pr\{[(\text{decide } H_1) \cap (H_1 = \text{true})] \cup \cdots \cup [(\text{decide } H_{\bar{n}}) \cap (H_{\bar{n}} = \text{true})]\} \tag{C.3.11}$$

The attribute "average" that has been attached to P_{cd} is motivated by the fact that (C.3.11) gives the probability of correct detection "averaged" over all possible hypotheses (as opposed, for example, to only considering the probability of correctly detecting that the model order is 2 (let us say), which is $\Pr\{\text{decide } H_2|H_2\}$).

Remark: Regarding the terminology, note that the determination of a real-valued parameter from the available data is called "estimation," whereas it is usually called "detection" for an integer-valued parameter, such as a model order. ■

In the following, we prove that *the MAP rule is optimal in the sense of maximizing P_{cd}*. To do so, consider a generic rule for selecting n, or, equivalently, for testing the hypotheses $\{H_n\}$ against each other. Such a rule will implicitly or explicitly partition the observation space, $\mathbf{R}^N$, into $\bar{n}$ sets $\{S_n\}_{n=1}^{\bar{n}}$, which are such that

$$\text{We decide } H_n \text{ if and only if } y \in S_n \tag{C.3.12}$$

Making use of (C.3.12) along with the fact that the hypotheses $\{H_n\}$ are mutually exclusive, we can write P_{cd} in (C.3.11) as

$$\begin{aligned} P_{cd} &= \sum_{n=1}^{\bar{n}} \Pr\{(\text{decide } H_n) \cap (H_n = \text{true})\} \\ &= \sum_{n=1}^{\bar{n}} \Pr\{(\text{decide } H_n)|H_n\} \Pr\{H_n\} \\ &= \sum_{n=1}^{\bar{n}} \int_{S_n} p_n(y|H_n) p_n(H_n)\, dy \\ &= \int_{\mathbf{R}^N} \left[\sum_{n=1}^{\bar{n}} I_n(y) p_n(y|H_n) p_n(H_n) \right] dy \end{aligned} \tag{C.3.13}$$

where $I_n(y)$ is the so-called indicator function, given by

$$I_n(y) = \begin{cases} 1, & \text{if } y \in S_n \\ 0, & \text{otherwise} \end{cases} \tag{C.3.14}$$

Next, observe that, for any given data vector, y, one and only one indicator function can be equal to 1 (because the sets S_n do not overlap, and their union is $\mathbf{R}^N$). This observation, along with the expression (C.3.13) for P_{cd}, implies that the MAP rule in (C.3.8) maximizes P_{cd}, as stated. Note that the sets $\{S_n\}$ corresponding to the MAP rule are implicitly defined via (C.3.8); however, $\{S_n\}$ are of no real interest in the proof, as both they and the indicator functions are introduced only to simplify the above proof. For more details on the topic of this subsection, we refer the reader to [Scharf 1991; Van Trees 1968].

C.3.2 Kullback–Leibler Information

Let $p_0(y)$ denote the *true pdf* of the observed data vector y, and let $\hat{p}(y)$ denote the pdf of a generic model of the data. The "discrepancy" between $p_0(y)$ and $\hat{p}(y)$ can be measured by using the Kullback–Leibler (KL) information or discrepancy function (see [Kullback and Leibler 1951]):

$$D(p_0, \hat{p}) = \int p_0(y) \ln\left[\frac{p_0(y)}{\hat{p}(y)}\right] dy \tag{C.3.15}$$

To simplify the notation, we omit the region of integration when it is the entire space. Letting $E_0\{\cdot\}$ denote the expectation with respect to the true pdf, $p_0(y)$, we can rewrite (C.3.15) as

$$D(p_0, \hat{p}) = E_0\left\{\ln\left[\frac{p_0(y)}{\hat{p}(y)}\right]\right\} = E_0\{\ln p_0(y)\} - E_0\left\{\ln \hat{p}(y)\right\} \tag{C.3.16}$$

Next, we prove that (C.3.15) possesses some properties of a suitable discrepancy function—namely,

$$\boxed{\begin{aligned} &D(p_0, \hat{p}) \geq 0 \\ &D(p_0, \hat{p}) = 0 \text{ if and only if } p_0(y) = \hat{p}(y) \end{aligned}} \tag{C.3.17}$$

To verify (C.3.17), we use the fact shown in Complement 6.5.8, that

$$-\ln\lambda \geq 1 - \lambda \quad \text{for any } \lambda > 0 \tag{C.3.18}$$

and

$$-\ln\lambda = 1 - \lambda \quad \text{if and only if } \lambda = 1 \tag{C.3.19}$$

Hence, letting $\lambda(y) = \hat{p}(y)/p_0(y)$, we have that

$$
\begin{aligned}
D(p_0, \hat{p}) &= \int p_0(y)\left[-\ln \lambda(y)\right] dy \\
&\geq \int p_0(y)\left[1 - \lambda(y)\right] dy = \int p_0(y)\left[1 - \frac{\hat{p}(y)}{p_0(y)}\right] dy = 0
\end{aligned}
$$

where the equality holds if and only if $\lambda(y) \equiv 1$, i.e. $\hat{p}(y) \equiv p_0(y)$.

Remark: The inequality in (C.3.17) also follows from Jensen's inequality (see equation (4.9.36) in Complement 4.9.5) and from the concavity of the function $\ln(\cdot)$:

$$
\begin{aligned}
D(p_0, \hat{p}) &= -E_0\left\{\ln\left[\frac{\hat{p}(y)}{p_0(y)}\right]\right\} \\
&\geq -\ln\left[E_0\left\{\frac{\hat{p}(y)}{p_0(y)}\right\}\right] \\
&= -\ln\left[\int \frac{\hat{p}(y)}{p_0(y)} p_0(y)\, dy\right] = -\ln(1) = 0 \qquad \blacksquare
\end{aligned}
$$

The KL discrepancy function can be viewed as quantifying the "loss of information" induced by the use of $\hat{p}(y)$ in lieu of $p_0(y)$. For this reason, $D(p_0, \hat{p})$ is sometimes called an information function, and the order-selection rules derived from it are called *information criteria* (see Sections C.4–C.6).

C.3.3 Outlook: Theoretical and Practical Perspectives

Neither the MAP rule nor the KL information can be used directly for order selection, because neither the pdfs of the data vector under the various hypotheses nor the true data pdf are available in any of the parametric spectral analysis problems discussed in the text. A possible way of using the MAP approach for order estimation consists of assuming an *a priori* pdf for the unknown parameter vector, θ^n, and integrating θ^n out of $p_n(y, \theta^n)$ to obtain $p_n(y|H_n)$. This Bayesian-type approach will be discussed in Section C.7. Regarding the KL approach, a natural way of using it for order selection consists in using an estimate, $\hat{D}(p_0, \hat{p})$, in lieu of the unavailable $D(p_0, \hat{p})$ (for a suitably chosen model pdf, $\hat{p}(y)$), and in determining the model order by minimizing $\hat{D}(p_0, \hat{p})$. This KL-based approach will be discussed in Sections C.4–C.6.

The derivations of all model order selection rules in the sections that follow rely on the assumption that one of the hypotheses $\{H_n\}$ is true. This assumption is unlikely to hold in applications with real-life data, so the reader will justifiably wonder whether an order-selection rule derived under such an assumption has any practical value. To address this concern, we remark that good parameter estimation methods (such as the MLM), derived under rather strict modeling assumptions, perform quite well in applications where the assumptions made are rarely satisfied exactly. Similarly, order-selection rules based on sound theoretical principles (such as the ML, KL, and MAP principles used in this text) are likely to perform well in applications despite the fact that some of the assumptions made when deriving them do not hold exactly. The precise behavior of order-selection rules (such as those presented in the sections to follow) in various mismodeling scenarios is not well understood, but extensive simulation results (see, e.g., [McQuarrie and Tsai 1998; Linhart and Zucchini 1986; Burnham and Anderson 2002]) lend support to this claim.

C.4 DIRECT KULLBACK–LEIBLER (KL) APPROACH: NO-NAME RULE

The model-dependent part of the Kullback–Leibler (KL) information, (C.3.16), is given by

$$-E_0\left\{\ln \hat{p}(y)\right\} \tag{C.4.1}$$

where $\hat{p}(y)$ is the pdf or likelihood of the model (to simplify the notation, we omit the index n of $\hat{p}(y)$; we will reinstate the index n later on, when needed). Minimization of (C.4.1) with respect to the model order is equivalent to *maximization* of the function

$$I(p_0, \hat{p}) \triangleq E_0\left\{\ln \hat{p}(y)\right\} \tag{C.4.2}$$

which is sometimes called the relative KL information. The ideal choice for $\hat{p}(y)$ in (C.4.2) would be the model likelihood, $p_n(y|H_n) = p_n(y, \theta^n)$. However, the model likelihood function is not available, and hence this choice is not possible. Instead, we might think of using

$$\hat{p}(y) = p(y, \hat{\theta}) \tag{C.4.3}$$

in (C.4.2), which would give

$$I\left(p_0, p(y, \hat{\theta})\right) = E_0\left\{\ln p(y, \hat{\theta})\right\} \tag{C.4.4}$$

Because the true pdf of the data vector is unknown, we cannot evaluate the expectation in (C.4.4). Apparently, what we could easily do is use the following unbiased estimate of $I\left(p_0, p(y, \hat{\theta})\right)$, instead of (C.4.4) itself:

$$\hat{I} = \ln p(y, \hat{\theta}) \tag{C.4.5}$$

However, the order-selection rule that maximizes (C.4.5) does *not* have satisfactory properties. This is especially true for *nested models*, in the case of which the order-selection rule based on the maximization of (C.4.5) *fails completely*: indeed, for nested models, this rule will always choose the maximum possible order, $\bar{n}$, because $\ln p_n(y, \hat{\theta}^n)$ increases monotonically with increasing n.

A better idea consists of approximating the unavailable log-pdf of the model, $\ln p_n(y, \theta^n)$, by a second-order Taylor series expansion around $\hat{\theta}^n$, and then using the approximation so obtained to define $\ln \hat{p}(y)$ in (C.4.2):

$$\begin{aligned} \ln p_n(y, \theta^n) \simeq{}& \ln p_n(y, \hat{\theta}^n) + (\theta^n - \hat{\theta}^n)^T \left[\left. \frac{\partial \ln p_n(y, \theta^n)}{\partial \theta^n} \right|_{\theta^n = \hat{\theta}^n} \right] \\ &+ \frac{1}{2}(\theta^n - \hat{\theta}^n)^T \left[\left. \frac{\partial^2 \ln p_n(y, \theta^n)}{(\partial \theta^n)\,(\partial \theta^n)^T} \right|_{\theta^n = \hat{\theta}^n} \right] (\theta^n - \hat{\theta}^n) \triangleq \ln \hat{p}_n(y) \end{aligned} \tag{C.4.6}$$

Because $\hat{\theta}^n$ is the maximizer of $\ln p_n(y, \theta^n)$, the second term in (C.4.6) is equal to zero. Hence, we can write (see also (C.2.21))

$$\ln \hat{p}_n(y) \simeq \ln p_n(y, \hat{\theta}^n) - \frac{1}{2}(\theta^n - \hat{\theta}^n)^T J(\theta^n - \hat{\theta}^n) \tag{C.4.7}$$

According to (C.2.11),

$$E_0\left\{(\theta^n - \hat{\theta}^n)^T J(\theta^n - \hat{\theta}^n)\right\} = \text{tr}\left[J E_0\left\{(\theta^n - \hat{\theta}^n)(\theta^n - \hat{\theta}^n)^T\right\}\right] = \text{tr}[I_n] = n \tag{C.4.8}$$

which means that, for the choice of $\hat{p}_n(y)$ in (C.4.7), we have

$$I = E_0\left\{\ln p_n(y, \hat{\theta}^n) - \frac{n}{2}\right\} \tag{C.4.9}$$

An unbiased estimate of the above relative KL information is given by

$$\ln p_n(y, \hat{\theta}^n) - \frac{n}{2} \tag{C.4.10}$$

The corresponding order-selection rule maximizes (C.4.10), or, equivalently, *minimizes*

$$\text{NN}(n) = -2\ln p_n(y, \hat{\theta}^n) + n \tag{C.4.11}$$

with respect to model order n. This no-name (NN) rule can be shown to perform better than that based on (C.4.5), but worse than the rules presented in the next sections. Essentially, the problem with (C.4.11) is that it tends to overfit (i.e., to select model orders larger than the “true” order). To understand intuitively how this happens, note that the first term in (C.4.11) decreases with increasing n (for nested models), whereas the second term increases. Hence, the second term in (C.4.11) *penalizes overfitting*; however, it turns out that it does not penalize quite enough. The rules presented in the following sections have a form similar to (C.4.11), but with a larger penalty term, and they do have better properties than (C.4.11). Despite this fact, we have chosen to present (C.4.11) briefly in this section for two reasons: (i) the discussion here has revealed the failure of using $\max_n \ln p_n(y, \hat{\theta}^n)$ as an order-selection rule *and* has shown that it is in effect quite easy to obtain rules with better properties; and (ii) this section has laid groundwork for the derivation of better order-selection rules based on the KL approach in the next two sections.

To close this section, we motivate the multiplication by -2 in going from (C.4.10) to (C.4.11). The reason for preferring (C.4.11) to (C.4.10) is that for the fairly common NLS model in (C.2.2) and the associated Gaussian likelihood in (C.2.3), $-2\ln p_n(y, \hat{\theta}^n)$ takes on the following convenient form:

$$-2\ln p_n(y, \hat{\theta}^n) = N \ln \hat{\sigma}_n^2 + \text{constant} \tag{C.4.12}$$

(See (C.2.5)–(C.2.7).) Hence, in such a case, we can replace $-2\ln p_n(y, \hat{\theta}^n)$ in (C.4.11) by the scaled logarithm of the residual variance, $N \ln \hat{\sigma}_n^2$. This remark also applies to the order-selection rules presented in the following sections, which are written in a form similar to (C.4.11).

C.5 CROSS-VALIDATORY KL APPROACH: THE AIC RULE

As explained in the previous section, a possible approach to model order selection consists of minimizing the KL discrepancy between the "true" pdf of the data and the pdf (or likelihood) of the model, or, equivalently, of maximizing the relative KL information (see (C.4.2)):

$$I(p_0, \hat{p}) = E_0\{\ln \hat{p}(y)\} \tag{C.5.1}$$

When using this approach, the first (and, likely the main) hurdle that we have to overcome is *the choice of the model likelihood*, $\hat{p}(y)$. As discussed in the previous section, we would ideally like to use the true pdf of the model as $\hat{p}(y)$ in (C.5.1), i.e. $\hat{p}(y) = p_n(y, \theta^n)$, but this is not possible; $p_n(y, \theta^n)$ is unknown. Hence, we have to choose $\hat{p}(y)$ in a different way. This choice is important; it eventually determines the model order selection rule that we will obtain. The other issue we should consider when using the approach based on (C.5.1) is that *the expectation in* (C.5.1) *cannot be evaluated*, because the true pdf of the data is unknown. Consequently, we will have to use an estimate, $\hat{I}$, in lieu of the unavailable $I(p_0, \hat{p})$ in (C.5.1).

Let x denote a *fictitious* data vector having the same size, N, and the same pdf as y, but such that x is *independent* of y. Also, let $\hat{\theta}_x$ denote the ML estimate of the model parameter vector that would be obtained from x if x were available. (We omit the superindex n of $\hat{\theta}_x$ as often as possible, to simplify notation.) In this section, we will consider the following choice of the model's pdf:

$$\ln \hat{p}(y) = E_x\left\{\ln p(y, \hat{\theta}_x)\right\} \tag{C.5.2}$$

which, when inserted in (C.5.1), yields

$$I = E_y\left\{E_x\left\{\ln p(y, \hat{\theta}_x)\right\}\right\} \tag{C.5.3}$$

Hereafter, $E_x\{\cdot\}$ and $E_y\{\cdot\}$ denote the expectation with respect to the pdf of x and y, respectively. The above choice of $\hat{p}(y)$, which was introduced in [Akaike 1974; Akaike 1978], has an interesting *cross-validation interpretation*: we use the sample x for estimation and the independent sample y for validation of the estimated model's pdf. Note that the dependence of (C.5.3) on the fictitious sample x is eliminated (as it should be, because x is unavailable) via the expectation operation $E_x\{\cdot\}$; see below for details.

An asymptotic second-order Taylor series expansion of $\ln p(y, \hat{\theta}_x)$ around $\hat{\theta}_y$, similar to (C.4.6)–(C.4.7), yields

$$\begin{aligned}
\ln p(y, \hat{\theta}_x) &\simeq \ln p(y, \hat{\theta}_y) + (\hat{\theta}_x - \hat{\theta}_y)^T \left[\left.\frac{\partial \ln p(y, \theta)}{\partial \theta}\right|_{\theta=\hat{\theta}_y} \right] \\
&\quad + \frac{1}{2}(\hat{\theta}_x - \hat{\theta}_y)^T \left[\left.\frac{\partial^2 \ln p(y, \theta)}{\partial \theta\, \partial \theta^T}\right|_{\theta=\hat{\theta}_y} \right] (\hat{\theta}_x - \hat{\theta}_y) \\
&\simeq \ln p(y, \hat{\theta}_y) - \frac{1}{2}(\hat{\theta}_x - \hat{\theta}_y)^T J_y (\hat{\theta}_x - \hat{\theta}_y)
\end{aligned} \tag{C.5.4}$$

where J_y is the J matrix, as defined in (C.2.20), associated with the data vector y. Using the fact that x and y have the same pdf (which implies that $J_y = J_x$), along with the fact that they are independent of each other, we can show that

$$\begin{aligned} & E_y\left\{E_x\left\{(\hat{\theta}_x - \hat{\theta}_y)^T J_y (\hat{\theta}_x - \hat{\theta}_y)\right\}\right\} \\ & = E_y\left\{E_x\left\{\mathrm{tr}\left(J_y\left[(\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta)\right]\left[(\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta)\right]^T\right)\right\}\right\} \\ & = \mathrm{tr}\left[J_y\left(J_x^{-1} + J_y^{-1}\right)\right] = 2n \end{aligned} \tag{C.5.5}$$

Inserting (C.5.5) in (C.5.4) yields the following asymptotic approximation of the relative KL information in (C.5.3):

$$I \simeq E_y\left\{\ln p_n(y, \hat{\theta}^n) - n\right\} \tag{C.5.6}$$

(where we have omitted the subindex y of $\hat{\theta}$ but reinstated the superindex n). Evidently, (C.5.6) can be estimated in an unbiased manner by

$$\ln p_n(y, \hat{\theta}^n) - n \tag{C.5.7}$$

Maximizing (C.5.7) with respect to n is equivalent to *minimizing* the function of n

$$\boxed{\mathrm{AIC} = -2\ln p_n(y, \hat{\theta}^n) + 2n} \tag{C.5.8}$$

where the acronym AIC stands for *Akaike Information Criterion* (the reasons for multiplying (C.5.7) by -2 to get (C.5.8), and for the use of the word "information" in the name given to (C.5.8) have been explained before—see the previous two sections).

As an example, for *the sinusoidal signal model* with n_c components (see Section C.2), AIC takes on the form (see (C.2.6)–(C.2.10))

$$\mathrm{AIC} = 2N_s \ln \hat{\sigma}_{n_c}^2 + 2(3n_c + 1) \tag{C.5.9}$$

where N_s denotes the number of available complex-valued samples, $\{y_c(t)\}_{t=1}^{N_s}$, and

$$\hat{\sigma}_{n_c}^2 = \frac{1}{N_s}\sum_{t=1}^{N_s}\left|y_c(t) - \sum_{k=1}^{n_c}\hat{\alpha}_k e^{i(\hat{\omega}_k t + \hat{\varphi}_k)}\right|^2 \tag{C.5.10}$$

Remark: AIC can also be obtained by using the following relative KL information function, in lieu of (C.5.3):

$$I = E_y\left\{E_x\left\{\ln p(x, \hat{\theta}_y)\right\}\right\} \tag{C.5.11}$$

Note that, in (C.5.11), x is used for validation and y for estimation. However, the derivation of AIC from (C.5.11) is more complicated; such a derivation, which is left as an exercise to the reader, makes use of two Taylor series expansions and of the fact that $E_x\{\ln p(x, \theta)\} = E_y\{\ln p(y, \theta)\}$.■

The performance of AIC has been found to be satisfactory in many case studies and applications to real-life data reported in the literature (see, for example, [McQuarrie and Tsai 1998; Linhart and Zucchini 1986; Burnham and Anderson 2002; Sakamoto, Ishiguro, and Kitagawa 1986]). *The performance of a model order selection rule*, such as AIC, can be measured in different ways, as explained in the next two paragraphs.

As a first possibility, we can consider a scenario in which the data-generating mechanism belongs to the class of models under test; thus, there is a true order. In such a case, analytical or numerical studies can be used to determine *the probability with which the rule selects the true order*. For AIC, it can be shown that, under quite general conditions,

$$\text{the probability of underfitting} \to 0 \tag{C.5.12}$$

$$\text{the probability of overfitting} \to \text{constant} > 0 \tag{C.5.13}$$

as $N \to \infty$ (see, for example, [McQuarrie and Tsai 1998; Kashyap 1980]). We can see from (C.5.13) that the behavior of AIC with respect to the probability of correct detection is not entirely satisfactory. Interestingly, it is precisely this kind of behavior that appears to make AIC perform satisfactorily with respect to the other possible type of performance measure, as explained below.

An alternative way of measuring the performance is to consider a more practical scenario, in which the data-generating mechanism is more complex than any of the models under test, as is usually the case in practical applications. In such a case, we can use analytical or numerical studies to determine the performance of the model picked by the rule as an *approximation* of the data-generating mechanism—for instance, we can consider the average distance between the estimated and true spectral densities or the average prediction error of the model. With respect to such a performance measure, AIC performs well, partly because of its tendency to select models with relatively large orders which may be a good thing to do in a case in which the data generating mechanism is more complex than the models used to fit it.

The nonzero overfitting probability of AIC is due to the fact that the term $2n$ in (C.5.8) (which penalizes high-order models), while larger than the term n that appears in the NN rule, is still too small. Extensive simulation studies (see, e.g., [Bhansali and Downham 1977]) have found empirically that the following Generalized Information Criterion (GIC)

$$\text{GIC} = -2 \ln p_n(y, \hat{\theta}^n) + \nu n \tag{C.5.14}$$

can outperform AIC with respect to various performance measures if $\nu > 2$. Specifically, depending on the considered scenario as well as the value of N and the performance measure, values of ν in the interval $\nu \in [2, 6]$ have been found to give the best performance.

In the next section, we show that GIC can be obtained as a natural theoretical extension of AIC. Hence, the use of (C.5.14) with $\nu > 2$ can be motivated on formal grounds. However, the choice of a particular ν in GIC is a more difficult problem, as we will see in Section C.6, and cannot be solved in the current KL framework. The different framework of Section C.7 appears to be necessary to arrive at a rule having the form of (C.5.14) with a specific expression for ν.

We close this section with a brief discussion on another modification of the AIC rule suggested in the literature (see, for example, [HURVICH AND TSAI 1993]). As explained before, AIC is derived by maximizing an *asymptotically* unbiased estimate of the relative KL information I in (C.5.3). Interestingly, for linear-regression models (given by (C.2.2) where $\mu(\gamma)$ is a linear function of γ), the following *corrected AIC rule*, AIC_c, can be shown to be an *exactly* unbiased estimate of I:

$$\text{AIC}_\text{c} = -2\ln p_n(y, \hat{\theta}^n) + \frac{2N}{N-n-1}n \tag{C.5.15}$$

(See, for example, [HURVICH AND TSAI 1993; CAVANAUGH 1997].) As $N \to \infty$, $\text{AIC}_\text{c} \to \text{AIC}$ (as expected). However, for finite values of N, the penalty term of AIC_c is larger than that of AIC. Consequently, in finite samples, AIC_c has a smaller risk of overfitting than AIC, and therefore we can say that AIC_c trades off a decrease of the risk of overfitting (which is rather large for AIC) for an increase in the risk of underfitting (which is quite small for AIC and hence can be slightly increased without a significant deterioration of performance). With this fact in mind, AIC_c can be used as an order-selection rule for models more general than just linear regressions, even though its motivation in the general case is pragmatic rather than theoretical. For other finite-sample corrections of AIC, we refer the reader to [DE WAELE AND BROERSEN 2003; BROERSEN 2000; BROERSEN 2002; SEGHOUANE, BEKARA, AND FLEURY 2003].

C.6 GENERALIZED CROSS-VALIDATORY KL APPROACH: THE GIC RULE

In the cross-validatory approach of the previous section, the estimation sample x has the same length as the validation sample y. In that approach, $\hat{\theta}_x$ (obtained from x) is used to approximate the likelihood of the model via $E_x\{p(y, \hat{\theta}_x)\}$. The AIC rule so obtained has a nonzero probability of overfitting (even asymptotically). Intuitively, the risk of overfitting will decrease if we let the length of the validation sample be (much) larger than that of the estimation sample—that is,

$$N \triangleq \text{length}(y) = \rho \cdot \text{length}(x), \qquad \rho \geq 1 \tag{C.6.1}$$

Indeed, overfitting occurs when the model corresponding to $\hat{\theta}_x$ also fits the "noise" in the sample x, so that $p(x, \hat{\theta}_x)$ has a "much" larger value than the true pdf, $p(x, \theta)$. Such a model could behave reasonably well on a short validation sample y, but not on a long validation sample. (In the latter case, $p(y, \hat{\theta}_x)$ will take on very small values.) The simple idea in (C.6.1) of letting the lengths of the validation and estimation samples be different leads to a natural extension of AIC, as shown next.

A straightforward calculation shows that, under (C.6.1), we have

$$J_y = \rho J_x \tag{C.6.2}$$

(See, e.g., (C.2.19).) With this small difference, the calculations in the previous section carry over to the present case, and we obtain (see (C.5.4)–(C.5.5))

$$\begin{aligned} I &\simeq E_y\left\{\ln p_n(y, \hat{\theta}_y)\right\} \\ &\quad - \frac{1}{2}E_y\left\{E_x\left\{\operatorname{tr}\left(J_y\left[(\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta)\right]\left[(\hat{\theta}_x - \theta) - (\hat{\theta}_y - \theta)\right]^T\right)\right\}\right\} \\ &= E_y\left\{\ln p_n(y, \hat{\theta}_y) - \frac{1}{2}\operatorname{tr}\left[J_y\left(\rho J_y^{-1} + J_y^{-1}\right)\right]\right\} \\ &= E_y\left\{\ln p_n(y, \hat{\theta}_y) - \frac{1+\rho}{2}n\right\} \end{aligned} \tag{C.6.3}$$

An unbiased estimate of the right side in (C.6.3) is given by

$$\ln p(y, \hat{\theta}_y) - \frac{1+\rho}{2}n \tag{C.6.4}$$

The *generalized information criterion (GIC) rule* maximizes (C.6.4) or, equivalently, *minimizes*

$$\boxed{\mathrm{GIC} = -2\ln p_n(y, \hat{\theta}^n) + (1+\rho)n} \tag{C.6.5}$$

As expected, (C.6.5) reduces to AIC for $\rho = 1$. Note also that, for a given y, the order selected by (C.6.5) with $\rho > 1$ is always smaller than the order selected by AIC (because the penalty term in (C.6.5) is larger than that in (C.5.8)); hence, as predicted by the previous intuitive discussion, the risk of overfitting associated with GIC is smaller than for AIC when $\rho > 1$.

On the negative side, there is no clear guideline for choosing ρ in (C.6.5). The "optimal" value of ρ in the GIC rule has been shown empirically to depend on the performance measure, the number of data samples, and the data-generating mechanism itself [McQuarrie and Tsai 1998; Bhansali and Downham 1977]. Consequently, ρ should be chosen as a function of all these factors, but there is no clear rule as to how that should be done. The approach of the next section appears to be more successful than the present approach in suggesting a specific choice for ρ in (C.6.5). Indeed, as we will see, that approach leads to an order-selection rule of the GIC type but with a concrete expression for ρ as a function of N.

C.7 BAYESIAN APPROACH: THE BIC RULE

The order-selection rule to be presented in this section can be obtained in two ways. First, let us consider *the KL framework* of the previous sections. Therefore, our goal is to maximize the relative KL information (see (C.5.1)):

$$I(p_0, \hat{p}) = E_0\left\{\ln \hat{p}(y)\right\} \tag{C.7.1}$$

The ideal choice of $\hat{p}(y)$ would be $\hat{p}(y) = p_n(y, \theta^n)$. However, this choice is not possible, because the likelihood of the model, $p_n(y, \theta^n)$, is not available. Hence, we have to use a "surrogate likelihood" in lieu of $p_n(y, \theta^n)$. Let us assume, as before, that a fictitious sample x is used to make inferences about θ. The pdf of the estimate, $\hat{\theta}_x$, obtained from x can alternatively be viewed as an *a priori* pdf of θ; hence, it will be denoted by $p(\theta)$ in what follows (once again, we omit the superindex n of θ, $\hat{\theta}$, *etc.* to simplify the notation, whenever there is no risk for confusion). Note that we do *not* constrain $p(\theta)$ to be Gaussian. We only assume that

$$p(\theta) \text{ is flat around } \hat{\theta} \tag{C.7.2}$$

where, as before, $\hat{\theta}$ denotes the ML estimate of the parameter vector obtained from the available data sample, y. Furthermore, now we assume that the length of the fictitious sample is a constant that does not depend on N; hence,

$$p(\theta) \text{ is independent of } N \tag{C.7.3}$$

As a consequence of assumption (C.7.3), the ratio between the lengths of the validation sample and the (fictitious) estimation sample grows without bound as N increases. According to the discussion in the previous section, this fact should lead to an order-selection rule having a penalty term asymptotically much larger than that of AIC or GIC (with $\rho =$ constant) and, hence, having a reduced risk of overfitting.

The scenario just introduced leads naturally to the following choice of surrogate likelihood:

$$\hat{p}(y) = E_\theta \left\{ p(y, \theta) \right\} = \int p(y, \theta) p(\theta)\, d\theta \tag{C.7.4}$$

Remark: In the previous sections, we used a surrogate likelihood given (see (C.5.2)) by

$$\ln \hat{p}(y) = E_x \left\{ \ln p(y, \hat{\theta}_x) \right\} \tag{C.7.5}$$

However, we could have instead used a $\hat{p}(y)$ given by

$$\hat{p}(y) = E_{\hat{\theta}_x} \left\{ p(y, \hat{\theta}_x) \right\} \tag{C.7.6}$$

The rule that would be obtained by using (C.7.6) can be shown to have the same form as AIC and GIC, but with a (slightly) different penalty term. Note that the choice of $\hat{p}(y)$ in (C.7.6) is similar to the choice in (C.7.4), with the difference that for (C.7.6) the "*a priori*" pdf, $p(\hat{\theta}_x)$, depends on N. ■

To obtain a simple asymptotic approximation of the integral in (C.7.4), we make use of the asymptotic approximation of $p(y, \theta)$ given by (C.4.6)–(C.4.7):

$$p(y, \theta) \simeq p(y, \hat{\theta}) e^{-\frac{1}{2}(\hat{\theta}-\theta)^T \hat{J} (\hat{\theta}-\theta)} \tag{C.7.7}$$

This equation holds for θ in the vicinity of $\hat{\theta}$. Inserting (C.7.7) in (C.7.4) and using the assumption in (C.7.2), along with the fact that $p(y, \theta)$ is asymptotically much larger at $\theta = \hat{\theta}$ than at any $\theta \neq \hat{\theta}$, we obtain

$$\begin{aligned}\hat{p}(y) &\simeq p(y, \hat{\theta})p(\hat{\theta}) \int e^{-\frac{1}{2}(\hat{\theta}-\theta)^T \hat{J} (\hat{\theta}-\theta)}\, d\theta \\ &= \frac{p(y, \hat{\theta})p(\hat{\theta})(2\pi)^{n/2}}{|\hat{J}|^{1/2}} \underbrace{\int \frac{1}{(2\pi)^{n/2}|\hat{J}^{-1}|^{1/2}} e^{-\frac{1}{2}(\hat{\theta}-\theta)^T \hat{J} (\hat{\theta}-\theta)}\, d\theta}_{=1} \\ &= \frac{p(y, \hat{\theta})p(\hat{\theta})(2\pi)^{n/2}}{|\hat{J}|^{1/2}} \end{aligned} \qquad \text{(C.7.8)}$$

(See [Djurić 1998] and references therein for the exact conditions under which this approximation holds true.) It follows from (C.7.1) and (C.7.8) that

$$\hat{I} = \ln p(y, \hat{\theta}) + \ln p(\hat{\theta}) + \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\hat{J}| \qquad \text{(C.7.9)}$$

is an asymptotically unbiased estimate of the relative KL information. Note, however, that (C.7.9) depends on the *a priori* pdf of θ, which has not been specified. To eliminate this dependence, we use the fact that $|\hat{J}|$ increases without bound as N increases. Specifically, in most cases (but not in all; see below) we have (*cf.* (C.2.21)) that

$$\ln |\hat{J}| = \ln \left| N \cdot \frac{1}{N} \hat{J} \right| = n \ln N + \ln \left| \frac{1}{N} \hat{J} \right| = n \ln N + \mathcal{O}(1) \qquad \text{(C.7.10)}$$

where we used the fact that $|cJ| = c^n |J|$ for a scalar c and an $n \times n$ matrix J. Using (C.7.10) and the fact that $p(\theta)$ is independent of N (see (C.7.3)) yields the following asymptotic approximation of the right side in (C.7.9):

$$\hat{I} \simeq \ln p_n(y, \hat{\theta}^n) - \frac{n}{2} \ln N \qquad \text{(C.7.11)}$$

The *Bayesian information criterion (BIC) rule* selects the order that maximizes (C.7.11), or, equivalently, *minimizes*

$$\boxed{\text{BIC} = -2 \ln p_n(y, \hat{\theta}^n) + n \ln N} \qquad \text{(C.7.12)}$$

We remind the reader that (C.7.12) has been derived under the assumption that (C.2.21) holds, but this assumption is *not* always true. As an example (see [Djurić 1998] for more examples), consider once again the sinusoidal signal model with n_c components (as also considered in Section C.5), in the case of which we have (*cf.* (C.2.22)–(C.2.23)) that

$$\begin{aligned}\ln|\hat{J}| &= \ln\left|K_N^{-2}\right| + \ln\left|K_N \hat{J} K_N\right| \\ &= (2n_c+1)\ln N_s + 3n_c \ln N_s + \mathcal{O}(1) \\ &= (5n_c+1)\ln N_s + \mathcal{O}(1) \end{aligned} \tag{C.7.13}$$

Hence, in the case of *sinusoidal signals*, BIC takes on the form

$$\begin{aligned}\text{BIC} &= -2\ln p_{n_c}(y, \hat{\theta}^{n_c}) + (5n_c+1)\ln N_s \\ &= 2N_s \ln \hat{\sigma}_{n_c}^2 + (5n_c+1)\ln N_s \end{aligned} \tag{C.7.14}$$

where $\hat{\sigma}_{n_c}^2$ is as defined in (C.5.10) and N_s denotes the number of complex-valued data samples.

The attribute Bayesian in the name of the rule in (C.7.12) or (C.7.14) is motivated by the use of the *a priori* pdf, $p(\theta)$, in the rule derivation, a method typical of a Bayesian approach. In fact, the BIC rule can be obtained by using a full Bayesian approach, as explained next.

To obtain the BIC rule in a *Bayesian framework*, we assume that the parameter vector θ is a random variable with a given *a priori* pdf denoted by $p(\theta)$. Owing to this assumption on θ, we need to modify the previously used notation as follows: $p(y,\theta)$ will now denote the joint pdf of y and θ, and $p(y|\theta)$ will denote the conditional pdf of y given θ. Using this notation and Bayes' rule, we can write

$$p(y|H_n) = \int p_n(y,\theta^n)\, d\theta^n = \int p_n(y|\theta^n) p_n(\theta^n)\, d\theta^n \tag{C.7.15}$$

The right side of (C.7.15) is identical to that of (C.7.4). It follows from this observation and from the analysis conducted in the first part of this section that, under the assumptions (C.7.2) and (C.7.3), and asymptotically in N,

$$\ln p(y|H_n) \simeq \ln p_n(y,\hat{\theta}^n) - \frac{n}{2}\ln N = -\frac{1}{2}\text{BIC} \tag{C.7.16}$$

Hence, maximizing $p(y|H_n)$ is asymptotically equivalent to minimizing BIC, independently of the prior $p(\theta)$ (as long as it satisfies (C.7.2) and (C.7.3)). The rediscovery of BIC in this Bayesian framework is important: It reveals the interesting fact that the BIC rule is asymptotically equivalent to the optimal MAP rule (see Section C.3.1) and, hence, that *the BIC rule can be expected to maximize the total probability of correct detection*, at least for sufficiently large values of N.

The BIC rule has been proposed in [SCHWARZ 1978A; KASHYAP 1982], among others. In [RISSANEN 1978; RISSANEN 1982] the same type of rule has been obtained by a different approach, one based on coding arguments and on the minimum description length (MDL) principle. The fact that the BIC rule can be derived in several different ways suggests that it might have a fundamental character. In particular, it can be shown that, under the assumption that the data-generating mechanism belongs to the model class considered, *the BIC rule is consistent*—that is,

$$\text{For BIC: the probability of correct detection} \to 1 \text{ as } N \to \infty \tag{C.7.17}$$

(See, e.g., [Söderström and Stoica 1989; McQuarrie and Tsai 1998].) This should be contrasted with the nonzero overfitting probability of AIC and GIC (with $\rho = \text{constant}$); see (C.5.12)–(C.5.13). Note that the result in (C.7.17) is not surprising in view of the asymptotic equivalence between the BIC rule and the optimal MAP rule.

Finally, we note in passing that, if we remove the condition in (C.7.3) that $p(\theta)$ be independent of N, then the term $\ln p(\hat{\theta})$ no longer may be eliminated from (C.7.9) by letting $N \to \infty$. Consequently, (C.7.9) would lead to a prior-dependent rule, which could be used to obtain any other rule described in this appendix by suitably choosing the prior. This line of argument can serve the theoretical purpose of interpreting various order-selection rules in a common Bayesian framework, but it appears to have little practical value; in particular, it can hardly be used to derive sound new order-selection rules.

C.8 SUMMARY AND THE MULTIMODEL APPROACH

In the first part of this section, we summarize the model order selection rules presented in the previous sections. Then we briefly discuss and motivate the multimodel approach which, as the name suggests, is based on the idea of using more than just one model for making inferences about the signal under study.

C.8.1 Summary

We begin with the observation that all the order-selection rules discussed in this appendix have the common form

$$-2\ln p_n(y, \hat{\theta}^n) + \eta(n, N)n \tag{C.8.1}$$

but different *penalty coefficients* $\eta(n, N)$:

$$\begin{aligned}
\text{AIC}: &\quad \eta(n, N) = 2 \\
\text{AIC}_\text{c}: &\quad \eta(n, N) = 2\frac{N}{N-n-1} \\
\text{GIC}: &\quad \eta(n, N) = \nu = \rho + 1 \\
\text{BIC}: &\quad \eta(n, N) = \ln N
\end{aligned} \tag{C.8.2}$$

Before using any of these rules for order selection in a specific problem, we need to carry out the following steps:

(i) Obtain an explicit expression for the term $-2\ln p_n(y, \hat{\theta}^n)$ in (C.8.1). This requires the specification both of the model structures to be tested and of their postulated likelihoods. An aspect that should receive some attention here is the fact that the derivation of all previous rules assumed real-valued data and parameters. Consequently, complex-valued data and parameters must be converted to real-valued quantities in order to apply the results in this appendix.

(ii) Count the number of unknown (real-valued) parameters in each model structure under consideration. This is easily done in the parametric spectral analysis problems in which we are interested.

(iii) Verify that the assumptions that have been made to derive the rules hold true. Fortunately, most of the assumptions made are quite weak; hence, they will usually hold. Indeed, the models under test may be either nested or non-nested, and they may even be only approximate descriptions of the data-generating mechanism. However, there are two particular assumptions, made on the information matrix J, that do not always hold and hence must be checked. First, we assumed in all derivations that the inverse matrix, J^{-1}, exists; such is not always the case. Second, we made the assumption that J is such that $J/N = \mathcal{O}(1)$. For some models, this is not true; when it is not true, a different normalization of J is required to make it tend to a constant matrix as $N \to \infty$. (This aspect is important for the BIC rule only.)

We have used the sinusoidal signal model as an example throughout this appendix to illustrate these steps and the involved aspects.

Once these aspects have been carefully considered, we can go on to use one of the four rules in (C.8.1)–(C.8.2) for selecting the order in our estimation problem. The question of which rule should be used is not an easy one. In general, we can prefer AIC_c over AIC: indeed, there is empirical evidence that AIC_c outperforms AIC in small samples (whereas in medium or large samples the two rules are almost equivalent). We also tend to prefer BIC over AIC or AIC_c, on the grounds that BIC is an asymptotic approximation of the optimal MAP rule. Regarding GIC, as mentioned in Sections C.5 and C.6, GIC with $\nu \in [2, 6]$ (depending on the scenario under study) can outperform AIC and AIC_c. Hence, for lack of a more precise guideline, we can think of using GIC with $\nu = 4$, the value in the middle of the above interval. To summarize, then, a possible ranking of the four rules discussed in this appendix is as follows (the first being considered the best):

- BIC
- GIC with $\nu = 4$ ($\rho = 3$)
- AIC_c
- AIC

In Figure C.1, we show the penalty coefficients of the above rules, as functions of N, to further illustrate the relationship between them.

C.8.2 The Multimodel Approach

We close this section with a brief discussion of a multimodel approach. Assume that we have used our favorite information criterion—call it XIC—and have computed its values for the model orders under test:

$$\text{XIC}(n); \quad n = 1, \ldots, \bar{n} \tag{C.8.3}$$

We can then pick the order that minimizes $\text{XIC}(n)$ and hence end up using a single model; this is the single-model approach.

Alternatively, we can consider a *multimodel approach*. Specifically, let us pick a $M \in [1, \bar{n}]$ (such as $M = 3$) and consider the model orders that give the M smallest values of $\text{XIC}(n)$, let

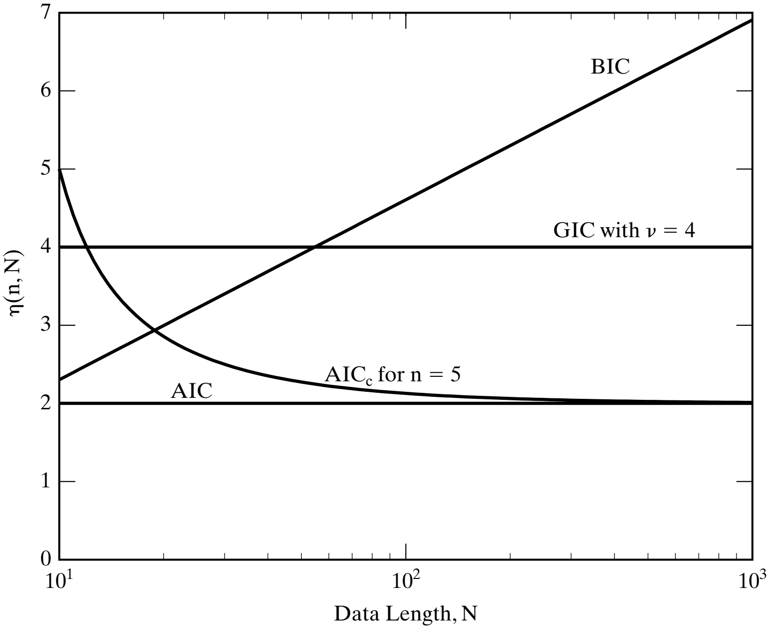


Figure C.1 Penalty coefficients of AIC, GIC with $\nu = 4$ ($\rho = 3$), AIC$_c$ (for $n = 5$), and BIC, as functions of data length N.

us say $n_1, \ldots, n_M$. From the derivations presented in the previous sections of this appendix, we can see that all information criteria attempt to estimate twice the negative log-likelihood of the model:

$$-2 \ln p_n(y, \theta^n) = -2 \ln p(y|H_n) \tag{C.8.4}$$

Hence, we can use

$$e^{-\frac{1}{2}\mathrm{XIC}(n)} \tag{C.8.5}$$

as an estimate of the likelihood of the model with order equal to n (to within a multiplicative constant). Consequently, instead of using just one model corresponding to the order that minimizes XIC(n), we can think of considering a combined use of the selected models (with orders $n_1, \ldots, n_M$) in which the contribution of each model is proportional to its likelihood value:

$$\frac{e^{-\frac{1}{2}\mathrm{XIC}(n_k)}}{\sum_{j=1}^{M} e^{-\frac{1}{2}\mathrm{XIC}(n_j)}}, \qquad k = 1, \ldots, M \tag{C.8.6}$$

For more details on the multimodel approach, including guidelines for choosing M, we refer the interested reader to [Burnham and Anderson 2002; Stoica, Selén, and Li 2004].

Appendix D

Answers to Selected Exercises

1.3(a): $\mathcal{Z}\{h_{-k}\} = H(1/z); \quad \mathcal{Z}\{g_k\} = H(z)H^*(1/z^*)$

1.4(a):

$$\begin{aligned}
\phi(\omega) &= \frac{\sigma^2}{(1 + a_1 e^{-i\omega})(1 + a_1^* e^{i\omega})} \left[1 + |b_1|^2 + b_1 e^{-i\omega} + b_1^* e^{i\omega}\right] \\
r(0) &= \frac{\sigma^2}{1 - |a_1|^2} \left\{|1 - b_1 a_1^*|^2 + |b_1|^2(1 - |a_1|^2)\right\} \\
r(k) &= \frac{\sigma^2}{1 - |a_1|^2} \left\{\left(1 - \frac{b_1}{a_1}\right)\left(1 - b_1^* a_1\right)\right\} (-a_1)^k, \qquad k \geq 1
\end{aligned}$$

1.9(a): $\phi_y(\omega) = \sigma_1^2 |H_1(\omega)|^2 + \rho\sigma_1\sigma_2 \left[H_1(\omega)H_2^*(\omega) + H_2(\omega)H_1^*(\omega)\right] + \sigma_2^2 |H_2(\omega)|^2$

2.3: An example is $y(t) = \{1, 1.1, 1\}$, whose unbiased ACS estimate is $\hat{r}(k) = \{1.07, 1.1, 1\}$, giving $\hat{\phi}(\omega) = 1.07 + 2.2\cos(\omega) + 2\cos(2\omega)$.

2.4(b): $\operatorname{var}\{\hat{r}(k)\} = \sigma^4 \alpha^2(k)(N - k)\left[1 + \delta_{k,0}\right]$

2.9:

(a) $E\left\{Y(\omega_k)Y^*(\omega_r)\right\} = \dfrac{\sigma^2}{N} \displaystyle\sum_{t=0}^{N-1} e^{i(\omega_r - \omega_k)t} = \begin{cases} \sigma^2 & k = r \\ 0 & k \neq r \end{cases}$

(c) $E\left\{\hat{\phi}(\omega)\right\} = \sigma^2 = \phi(\omega)$, so $\hat{\phi}(\omega)$ is an unbiased estimate.

3.2: Decompose the ARMA system as $x(t) = \frac{1}{A(z)}e(t)$ and $y(t) = B(z)x(t)$. Then $\{x(t)\}$ is an AR(n) process. To find $\{r_x(k)\}$ from $\{\sigma^2, a_1 \dots a_n\}$, write the Yule–Walker equations as

$$\begin{bmatrix} 1 & & 0 \\ a_1 & \ddots & \\ \vdots & & \ddots \\ a_n & \dots \; a_1 & 1 \end{bmatrix} \begin{bmatrix} r_x(0) \\ r_x(1) \\ \vdots \\ r_x(n) \end{bmatrix} + \begin{bmatrix} 0 & a_1 & \dots & a_n \\ \vdots & \vdots & & 0 \\ \vdots & a_n & 0 & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} r_x^*(0) \\ r_x^*(1) \\ \vdots \\ r_x^*(n) \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or

$$A_1 r_x + A_2 r_x^c = \begin{bmatrix} \sigma^2 \\ 0 \end{bmatrix}$$

which can be solved for $\{r_x(m)\}_{m=0}^n$. Then find $r_x(k)$ for $k > n$ from equation (3.3.4) and $r_x(k)$ for $k < 0$ from $r_x^*(-k)$. Finally,

$$r_y(k) = \sum_{j=0}^{m} \sum_{p=0}^{m} r_x(k + p - j)\, b_j b_p^*$$

3.4: $\sigma_b^2 = E\left\{|e_b(t)|^2\right\} = [1 \; \theta_b^T] R_{n+1} \begin{bmatrix} 1 \\ \theta_b^c \end{bmatrix} = [1 \; \theta_b^*] R_{n+1}^c \begin{bmatrix} 1 \\ \theta_b \end{bmatrix}$ giving $\theta_b = \theta_f$ and $\sigma_b^2 = \sigma_f^2$.

3.5(a):

$$R_{2m+1}^T \begin{bmatrix} c_m \\ \vdots \\ c_1 \\ 1 \\ d_1 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma_s^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

3.14: $c_\ell = \sum_{i=0}^n a_i \tilde{r}(\ell - i)$ for $0 \le \ell \le p$, where $\tilde{r}(k) = r(k)$ for $k \ge 1$, $\tilde{r}(0) = r(0)/2$, and $\tilde{r}(k) = 0$ for $k < 0$.

3.15(b): First solve for $b_1, \dots, b_m$ from

$$\begin{bmatrix} c_n & c_{n-1} & \cdots & c_{n-m+1} \\ c_{n+1} & c_n & \cdots & c_{n-m+2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n+m-1} & c_{n+m-2} & \dots & c_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = - \begin{bmatrix} c_{n+1} \\ c_{n+2} \\ \vdots \\ c_{n+m} \end{bmatrix}$$

Then $a_1, \dots, a_n$ can be obtained from $a_k = c_k + \sum_{i=1}^m b_i c_{k-i}$.

4.2:
(a) $E\{x(t)\} = 0$; $r_x(k) = (\overline{\alpha}^2 + \sigma_\alpha^2)e^{i\omega_0 k}$
(b) Let $p(\varphi) = \sum_{k=-\infty}^{\infty} c_k e^{-ik\varphi}$ be the Fourier series of $p(\varphi)$ for $\varphi \in [-\pi, \pi]$. Then $E\{x(t)\} = \frac{\overline{\alpha} e^{i\omega_0 t}}{2\pi} c_1$. Thus, $E\{x(t)\} = 0$ if and only if either $\overline{\alpha} = 0$ or $c_1 = 0$. In this case, $r_x(k)$ is the same as in part (a).

5.8: The height of the peak of the (unnormalized) Capon spectrum is

$$1/a^*(\omega)R^{-1}a(\omega)|_{\omega=\omega_0} = \frac{m\alpha^2 + \sigma^2}{m}$$

1

Basic Concepts

1.1 INTRODUCTION

The essence of the spectral estimation problem is captured by the following informal formulation.

> From a finite record of a stationary data sequence, estimate how the total power is distributed over frequency. (1.1.1)

Spectral analysis finds applications in many diverse fields. In *vibration monitoring*, the spectral content of measured signals gives information on the wear and other characteristics of mechanical parts under study. In *economics*, *meteorology*, *astronomy*, and several other fields, the spectral analysis may reveal “hidden periodicities” in the studied data, which are to be associated with cyclic behavior or recurring processes. In *speech analysis*, spectral models of voice signals are useful in better understanding the speech production process and—in addition—can be used for both speech synthesis (or compression) and speech recognition. In *radar and sonar systems*, the spectral contents of the received signals provide information on the location of the sources (or targets) situated in the field of view. In *medicine*, spectral analysis of various signals measured from a patient, such as electrocardiogram (ECG) or electroencephalogram (EEG) signals, can provide useful material for diagnosis. In *seismology*, the spectral analysis of the signals recorded prior to and during a seismic event (such as a volcano eruption or an earthquake) gives useful information on the ground movement associated with such events and could help in predicting them. Seismic spectral estimation is also used to predict subsurface geologic structure in gas and oil exploration. In *control systems*, there is a resurging interest in spectral analysis methods as a means of characterizing the dynamical behavior of a given system and ultimately synthesizing a controller for that system. The previous and other applications of spectral analysis are reviewed in [KAY 1988; MARPLE 1987; BLOOMFIELD 1976; BRACEWELL 1986; HAYKIN 1991; HAYKIN 1995; HAYES III 1996; KOOPMANS 1974; PRIESTLEY 1981; PERCIVAL AND WALDEN 1993; PORAT 1994; SCHARF 1991; THERRIEN 1992; PROAKIS, RADER, LING, AND NIKIAS 1992]. The textbook [MARPLE 1987] also contains a well-written historical perspective on spectral estimation, which is worth reading. Many of the classical articles on spectral analysis, both application-driven and theoretical, are reprinted in [CHILDERS 1978; KESLER 1986]; these excellent collections of reprints are well worth consulting.

There are *two broad approaches* to spectral analysis. One of these derives its basic idea directly from definition (1.1.1): The studied signal is applied to a bandpass filter with a narrow bandwidth, which is swept through the frequency band of interest, and the filter output power divided by the filter bandwidth is used as a measure of the spectral content of the input to the filter. This is essentially what the *classical* (or *nonparametric*) *methods* of spectral analysis do. These methods are described in Chapters 2 and 5 of this text. (The fact that the methods of Chapter 2 can be given the filter-bank interpretation is made clear in Chapter 5.) The second approach to spectral estimation, called the *parametric approach*, is to postulate a model for the data, which provides a means of parameterizing the spectrum, and to thereby reduce the spectral estimation problem to that of estimating the parameters in the assumed model. The parametric approach to spectral analysis is treated in Chapters 3, 4, and 6. Parametric methods offer more accurate spectral estimates than the nonparametric ones in the cases where the data indeed satisfy the model assumed by the former methods. However, in the more likely case that the data do not satisfy the assumed models, the nonparametric methods sometimes outperform the parametric ones, because of the sensitivity of the latter to model misspecifications. This observation has motivated renewed interest in the nonparametric approach to spectral estimation.

Many real-world signals can be characterized as being *random* (from the observer's viewpoint). Briefly speaking, this means that the variation of such a signal outside the observed interval cannot be determined exactly, but can only be specified in statistical terms of averages. In this text, we will be concerned with estimating the spectral characteristics of random signals. In spite of this fact, we find it useful to start the discussion by considering the spectral analysis of deterministic signals (as we do in Section 1.2). Throughout this work, we consider *discrete-index signals* (or *data sequences*). Such signals are most commonly obtained by the temporal or spatial sampling of a continuous (in time or space) signal. The main motivation for focusing on discrete signals lies in the fact that spectral analysis is most often performed by a digital computer or by digital circuitry. Chapters 2 to 5 of this text deal with *discrete-time signals*; Chapter 6 considers the case of *discrete-space data sequences*.

In the interest of notational simplicity, the discrete-time variable t, as used in this text, is assumed to be measured in units of sampling interval. A similar convention is adopted for spatial signals, whenever the sampling is uniform. Accordingly, the *units of frequency* are cycles per sampling interval.

The signals dealt with in the text are *complex valued*. Complex-valued data can appear in signal processing and spectral estimation applications—for instance, as a result of a "complex demodulation" process (explained in detail in Chapter 6). It should be noted that the treatment of complex-valued signals is not always more general or more difficult than the analysis of corresponding real-valued signals. A typical example that illustrates this claim is the case of sinusoidal signals considered in Chapter 4. A real-valued sinusoidal signal, $\alpha \cos(\omega t + \varphi)$, can be rewritten as a linear combination of two complex-valued sinusoidal signals, $\alpha_1 e^{i(\omega_1 t + \varphi_1)} + \alpha_2 e^{i(\omega_2 t + \varphi_2)}$, whose parameters are constrained as follows: $\alpha_1 = \alpha_2 = \alpha/2$, $\varphi_1 = -\varphi_2 = \varphi$, and $\omega_1 = -\omega_2 = \omega$. Here $i = \sqrt{-1}$. The fact that we need to consider *two constrained* complex sine waves to treat the case of *one unconstrained* real sine wave shows that the real-valued case of sinusoidal signals can actually be considered to be more complicated than the complex-valued case! Fortunately, it appears that the latter case is encountered more frequently in applications, where often both the *in-phase* and *quadrature* components of the studied signal are available. For more details and explanations on this aspect, see Section 6.2.

1.2 ENERGY SPECTRAL DENSITY OF DETERMINISTIC SIGNALS

Let $\{y(t); t = 0, \pm 1, \pm 2, \ldots\}$ denote a *deterministic* discrete-time data sequence. Most commonly, $\{y(t)\}$ is obtained by sampling a continuous-time signal. For notational convenience, the time index t is expressed in units of sampling interval—that is, $y(t) = y_c(t \cdot T_s)$, where $y_c(\cdot)$ is the continuous time signal and T_s is the sampling time interval.

Assume that $\{y(t)\}$ has *finite energy*, which means that

$$\sum_{t=-\infty}^{\infty} |y(t)|^2 < \infty \tag{1.2.1}$$

Then, under some additional regularity conditions, the sequence $\{y(t)\}$ possesses a *discrete-time Fourier transform* (DTFT) defined as

$$\boxed{Y(\omega) = \sum_{t=-\infty}^{\infty} y(t) e^{-i\omega t} \qquad \text{(DTFT)}} \tag{1.2.2}$$

In this text, we use the symbol $Y(\omega)$, in lieu of the more cumbersome $Y(e^{i\omega})$, to denote the DTFT. This notational convention is commented on a bit later, following equation (1.4.6). The corresponding inverse DTFT is then

$$\boxed{y(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\omega) e^{i\omega t} d\omega \qquad \text{(Inverse DTFT)}} \tag{1.2.3}$$

which can be verified by substituting (1.2.3) into (1.2.2). The (angular) *frequency* ω is measured in radians per sampling interval. The conversion from ω to the *physical frequency variable* $\bar{\omega} = \omega / T_s$ [rad/sec] can be done in a straightforward manner, as described in Exercise 1.1.

Let

$$\boxed{S(\omega) = |Y(\omega)|^2 \qquad \text{(Energy Spectral Density)}} \tag{1.2.4}$$

A straightforward calculation gives

$$\begin{aligned}\frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)d\omega &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{t=-\infty}^{\infty}\sum_{s=-\infty}^{\infty} y(t)y^*(s)e^{-i\omega(t-s)}d\omega \\ &= \sum_{t=-\infty}^{\infty}\sum_{s=-\infty}^{\infty} y(t)y^*(s)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-i\omega(t-s)}d\omega\right] \\ &= \sum_{t=-\infty}^{\infty} |y(t)|^2 \qquad (1.2.5)\end{aligned}$$

To obtain the last equality in (1.2.5), we have used the fact that $\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-i\omega(t-s)}d\omega = \delta_{t,s}$ (the Kronecker delta). The symbol $(\cdot)^*$ will be used in this text to denote the complex conjugate of a scalar variable or the conjugate transpose of a vector or matrix. Equation (1.2.5) can be restated as

$$\boxed{\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)d\omega} \qquad (1.2.6)$$

This equality is called *Parseval's theorem*. It shows that $S(\omega)$ represents the distribution of sequence energy as a function of frequency. For this reason, $S(\omega)$ is called the *energy spectral density*.

The previous interpretation of $S(\omega)$ also comes up in the following way: Equation (1.2.3) represents the sequence $\{y(t)\}$ as a weighted "sum" (actually, an integral) of orthonormal sequences $\{\frac{1}{\sqrt{2\pi}}e^{i\omega t}\}$ ($\omega \in [-\pi, \pi]$), with weighting $\frac{1}{\sqrt{2\pi}}Y(\omega)$. Hence, $\frac{1}{\sqrt{2\pi}}|Y(\omega)|$ "measures" the "length" of the projection of $\{y(t)\}$ on each of these basis sequences. In loose terms, therefore, $\frac{1}{\sqrt{2\pi}}|Y(\omega)|$ shows how much (or how little) of the sequence $\{y(t)\}$ can be "explained" by the orthonormal sequence $\{\frac{1}{\sqrt{2\pi}}e^{i\omega t}\}$ for some given value of ω.

Define

$$\rho(k) = \sum_{t=-\infty}^{\infty} y(t)y^*(t-k) \qquad (1.2.7)$$

It is readily verified that

$$\begin{aligned}\sum_{k=-\infty}^{\infty}\rho(k)e^{-i\omega k} &= \sum_{k=-\infty}^{\infty}\sum_{t=-\infty}^{\infty} y(t)y^*(t-k)e^{-i\omega t}e^{i\omega(t-k)} \\ &= \left[\sum_{t=-\infty}^{\infty} y(t)e^{-i\omega t}\right]\left[\sum_{s=-\infty}^{\infty} y(s)e^{-i\omega s}\right]^* \\ &= S(\omega) \qquad (1.2.8)\end{aligned}$$

which shows that $S(\omega)$ can be obtained as the DTFT of the "autocorrelation" (1.2.7) of the finite-energy sequence $\{y(t)\}$.

These definitions can be extended in a rather straightforward manner to the case of random signals treated throughout the remaining text. In fact, the only purpose for discussing the deterministic case in this section was to provide some motivation for the analogous definitions in the random case. As such, the discussion in this section has been kept brief. More insights into the meaning and properties of the previous definitions are provided by the detailed treatment of the random case in the next sections.

1.3 POWER SPECTRAL DENSITY OF RANDOM SIGNALS

Most of the signals encountered in applications are such that their future values cannot be determined exactly. We thus resort to probabilistic statements about future values. The mathematical device to describe such a signal is that of a *random sequence*, which consists of an ensemble of possible realizations, each of which has some associated probability of occurrence. Of course, from the whole ensemble of realizations, the experimenter can usually observe only one realization of the signal, and then it might be thought that the deterministic definitions of the previous section could be carried over unchanged to the present case. However, this is not possible, because the realizations of a random signal, viewed as discrete-time sequences, do not have finite energy and hence do not possess DTFTs. A random signal usually has finite *average* power and, therefore, can be characterized by an average power spectral density. For simplicity reasons, in what follows we will use the name *power spectral density* (PSD) for that quantity.

The discrete-time signal $\{y(t); t = 0, \pm1, \pm2, \ldots\}$ is assumed to be a sequence of random variables with *zero mean*:

$$E\{y(t)\} = 0 \qquad \text{for all} \quad t \tag{1.3.1}$$

Hereafter, $E\{\cdot\}$ denotes the expectation operator (which averages over the ensemble of realizations). The *autocovariance sequence* (ACS) or *covariance function* of $y(t)$ is defined as

$$\boxed{r(k) = E\left\{y(t)y^*(t-k)\right\}} \tag{1.3.2}$$

and it is assumed to depend only on the lag between the two samples averaged. The two assumptions (1.3.1) and (1.3.2) imply that $\{y(t)\}$ is a *second-order stationary sequence*. When it is required to distinguish between the autocovariance sequences of several signals, a lower index will be used to indicate the signal associated with a given covariance lag, such as $r_y(k)$.

The autocovariance sequence $r(k)$ enjoys some simple, but useful, properties:

$$\boxed{r(k) = r^*(-k)} \tag{1.3.3}$$

and

$$\boxed{r(0) \geq |r(k)| \qquad \text{for all } k} \tag{1.3.4}$$

The equality (1.3.3) directly follows from definition (1.3.2) and the stationarity assumption; (1.3.4) is a consequence of the fact that the *covariance matrix* of $\{y(t)\}$, defined as follows:

$$R_m = \begin{bmatrix} r(0) & r^*(1) & \dots & r^*(m-1) \\ r(1) & r(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & r^*(1) \\ r(m-1) & \dots & r(1) & r(0) \end{bmatrix}$$

$$= E\left\{ \begin{bmatrix} y^*(t-1) \\ \vdots \\ \vdots \\ y^*(t-m) \end{bmatrix} [y(t-1)\dots y(t-m)] \right\} \tag{1.3.5}$$

is positive semidefinite for all m. Recall that a Hermitian matrix M is positive semidefinite if $a^*Ma \geq 0$ for every vector a; see Section A.5 for details. Now,

$$a^*R_m a = a^*E\left\{ \begin{bmatrix} y^*(t-1) \\ \vdots \\ y^*(t-m) \end{bmatrix} [y(t-1)\dots y(t-m)] \right\} a$$

$$= E\left\{z^*(t)z(t)\right\} = E\left\{|z(t)|^2\right\} \geq 0 \tag{1.3.6}$$

where

$$z(t) = [y(t-1)\dots y(t-m)]a$$

so we see that R_m is indeed positive semidefinite for every m. Hence, (1.3.4) follows from the properties of positive semidefinite matrices. (See Definition D11 in Appendix A and Exercise 1.5.)

1.3.1 First Definition of Power Spectral Density

The PSD is defined as the DTFT of the covariance sequence:

$$\boxed{\phi(\omega) = \sum_{k=-\infty}^{\infty} r(k)e^{-i\omega k} \qquad \text{(Power Spectral Density)}} \tag{1.3.7}$$

Note that the previous definition (1.3.7) of $\phi(\omega)$ is similar to the definition (1.2.8) in the deterministic case. The inverse transform, which recovers $\{r(k)\}$ from a given $\phi(\omega)$, is

$$\boxed{r(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi(\omega)e^{i\omega k}d\omega} \tag{1.3.8}$$

We readily verify that

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \phi(\omega)e^{i\omega k}d\omega = \sum_{p=-\infty}^{\infty} r(p)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{i\omega(k-p)}d\omega\right] = r(k)$$

which proves that (1.3.8) is the inverse transform for (1.3.7). Note that, to obtain the first equality described, the order of integration and summation has been inverted. This order inversion is possible under weak conditions, such as when $\phi(\omega)$ is square integrable—see Chapter 4 in [Priestley 1981] for a detailed discussion on this aspect.

From (1.3.8), we obtain

$$r(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi(\omega)d\omega \tag{1.3.9}$$

Since $r(0) = E\left\{|y(t)|^2\right\}$ measures the (average) power of $\{y(t)\}$, the equality (1.3.9) shows that $\phi(\omega)$ can indeed be named PSD, as it represents the distribution of the (average) signal power over frequencies. Put another way, it follows from (1.3.9) that $\phi(\omega)d\omega/2\pi$ is the infinitesimal power in the band $(\omega - d\omega/2,\ \omega + d\omega/2)$, and that the total power in the signal is obtained by integrating these infinitesimal contributions. Additional motivation for calling $\phi(\omega)$ a PSD is provided by the second definition of $\phi(\omega)$, given next, which resembles the usual definition (1.2.2), (1.2.4) in the deterministic case.

1.3.2 Second Definition of Power Spectral Density

The second definition of $\phi(\omega)$ is

$$\boxed{\phi(\omega) = \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\omega t}\right|^2\right\}} \tag{1.3.10}$$

This definition is equivalent to (1.3.7) under the mild assumption that the covariance sequence $\{r(k)\}$ decays sufficiently rapidly that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=-N}^{N} |k||r(k)| = 0 \tag{1.3.11}$$

The equivalence of (1.3.7) and (1.3.10) can be verified as follows:

$$\begin{aligned}
\lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\omega t}\right|^2\right\} &= \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\sum_{s=1}^{N} E\left\{y(t)y^*(s)\right\} e^{-i\omega(t-s)} \\
&= \lim_{N\to\infty}\frac{1}{N}\sum_{\tau=-(N-1)}^{N-1} (N-|\tau|)r(\tau)e^{-i\omega\tau} \\
&= \sum_{\tau=-\infty}^{\infty} r(\tau)e^{-i\omega\tau} - \lim_{N\to\infty}\frac{1}{N}\sum_{\tau=-(N-1)}^{N-1} |\tau| r(\tau)e^{-i\omega\tau} \\
&= \phi(\omega)
\end{aligned}$$

The second equality is proven in Exercise 1.6, and we used (1.3.11) in the last equality.

The second definition just mentioned of $\phi(\omega)$ resembles the definition (1.2.4) of energy spectral density in the deterministic case. The main difference between (1.2.4) and (1.3.10) consists of the appearance of the expectation operator in (1.3.10) and the normalization by $1/N$; the fact that the "discrete-time" variable in (1.3.10) runs over positive integers only is just for convenience and does not constitute an essential difference, compared with (1.2.2). In spite of these differences, the analogy between the deterministic formula (1.2.4) and (1.3.10) provides further motivation for calling $\phi(\omega)$ a PSD. The alternative definition (1.3.10) will also be quite useful when discussing the problem of estimating the PSD by nonparametric techniques in Chapters 2 and 5.

We can see, from either of these definitions, that $\phi(\omega)$ is a *periodic function*, with the period equal to 2π. Hence, $\phi(\omega)$ is completely described by its variation in the interval

$$\boxed{\omega \in [-\pi, \pi] \qquad \text{(radians per sampling interval)}} \tag{1.3.12}$$

Alternatively, the PSD can be viewed as a function of the frequency

$$\boxed{f = \frac{\omega}{2\pi} \qquad \text{(cycles per sampling interval)}} \tag{1.3.13}$$

which, according to (1.3.12), can be considered to take values in the interval

$$\boxed{f \in [-1/2, 1/2]} \tag{1.3.14}$$

We will generally write the PSD as a function of ω whenever possible, because doing so will simplify the notation.

As already mentioned, the discrete-time sequence $\{y(t)\}$ is most commonly derived by sampling a continuous-time signal. To avoid aliasing effects that might be incurred by the sampling process, the continuous-time signal should be (at least, approximately) bandlimited in the frequency domain. To ensure this, it may be necessary to lowpass filter the continuous-time signal before sampling. Let F_0 denote the largest ("significant") frequency component in the spectrum of the (possibly filtered) continuous signal, and let F_s be the *sampling frequency*. Then it follows from the Nyquist sampling theorem (sometimes called the Whittaker–Nyquist–Kotelnikov–Shannon sampling theorem) that the continuous-time signal can be exactly reconstructed from its samples $\{y(t)\}$, provided that

$$F_s > 2F_0 \tag{1.3.15}$$

In particular, no frequency aliasing will occur when (1.3.15) holds. (See, for example, [Oppenheim and Schafer 1989].) The frequency variable, F, associated with the continuous-time signal is related to f by the equation

$$F = f \cdot F_s \tag{1.3.16}$$

so it follows that the interval of F corresponding to (1.3.14) is

$$\boxed{F \in \left[-\frac{F_s}{2} \,,\, \frac{F_s}{2} \right] \qquad \text{(cycles/sec)}} \tag{1.3.17}$$

which is quite natural in view of (1.3.15).

1.4 PROPERTIES OF POWER SPECTRAL DENSITIES

Since $\phi(\omega)$ is a power density, it should be real valued and nonnegative. That this is indeed the case is readily seen from definition (1.3.10) of $\phi(\omega)$. Hence,

$$\boxed{\phi(\omega) \geq 0 \qquad \text{for all} \;\; \omega} \tag{1.4.1}$$

From (1.3.3) and (1.3.7), we obtain

$$\phi(\omega) = r(0) + 2\sum_{k=1}^{\infty} \operatorname{Re}\{r(k)e^{-i\omega k}\}$$

where $\operatorname{Re}\{\cdot\}$ denotes the real part of the bracketed quantity. If $y(t)$, and hence $r(k)$, is real valued, then it follows that

$$\phi(\omega) = r(0) + 2\sum_{k=1}^{\infty} r(k)\cos(\omega k) \tag{1.4.2}$$

which shows that $\phi(\omega)$ is an even function in such a case. In the case of complex-valued signals, however, $\phi(\omega)$ is not necessarily symmetric about the $\omega = 0$ axis. Thus,

$$\boxed{\begin{array}{l}\text{For real-valued signals:}\\ \qquad\qquad\qquad \phi(\omega) = \phi(-\omega), \ \ \omega \in [-\pi, \pi] \\[1ex] \text{For complex-valued signals:}\\ \qquad\qquad\qquad \text{in general } \phi(\omega) \neq \phi(-\omega), \ \ \omega \in [-\pi, \pi]\end{array}} \tag{1.4.3}$$

Remark: The reader might wonder why we did not define the ACS as

$$c(k) = E\left\{y(t)y^*(t+k)\right\}$$

Comparing with the ACS $\{r(k)\}$ used in this text, as defined in (1.3.2), we obtain $c(k) = r(-k)$. Consequently, the PSD associated with $\{c(k)\}$ is related to the PSD corresponding to $\{r(k)\}$ (see (1.3.7)) via

$$\psi(\omega) \triangleq \sum_{k=-\infty}^{\infty} c(k)e^{-i\omega k} = \sum_{k=-\infty}^{\infty} r(k)e^{i\omega k} = \phi(-\omega)$$

It may seem arbitrary as to which definition of the ACS (and corresponding definition of PSD) we choose. In fact, from a mathematical standpoint we can use either definition of the ACS, but the ACS definition $r(k)$ is preferred from a practical standpoint, as we now explain.

First, we should stress that the PSD describes the spectral content of the ACS, as seen from equation (1.3.7). The PSD $\phi(\omega)$ is sometimes perceived as showing the (infinitesimal) power at frequency ω in the signal itself, but that is not strictly true. If the PSD represented the power in the signal itself, then we should have had $\psi(\omega) = \phi(\omega)$, because the signal's spectral content should not depend on the ACS definition. However, as shown earlier, in the general complex case, $\psi(\omega) = \phi(-\omega) \neq \phi(\omega)$, which means that the signal power interpretation of the PSD is not (always) correct. Indeed, the PSD $\phi(\omega)$ "measures" the *power at frequency ω in the signal's ACS*.

On the other hand, our motivation for considering spectral analysis is to characterize the *average power* at frequency *ω in the signal*, as given by the second definition of the PSD in equation (1.3.10). If $c(k)$ is used as the ACS, its corresponding second definition of the PSD is

$$\psi(\omega) = \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{+i\omega t}\right|^2\right\}$$

which is the average power of $y(t)$ at frequency $-\omega$. Clearly, the second PSD definition corresponding to $r(k)$ aligns with this average-power motivation, whereas the one for $c(k)$ does not; it is for this reason that we use the definition $r(k)$ for the ACS. ■

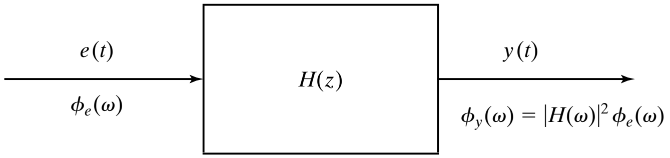


Figure 1.1 Relationship between the PSDs of the input and output of a linear system.

Next, we present a useful result that concerns the *transfer of PSD through an asymptotically stable linear system.* Let

$$H(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k} \tag{1.4.4}$$

denote an asymptotically stable linear time-invariant system. The symbol z^{-1} denotes the unit delay operator defined by $z^{-1}y(t) = y(t-1)$. Also, let $e(t)$ be the stationary input to the system and $y(t)$ the corresponding output, as shown in Figure 1.1.Then $\{y(t)\}$ and $\{e(t)\}$ are related via the convolution sum

$$y(t) = H(z)e(t) = \sum_{k=-\infty}^{\infty} h_k e(t-k) \tag{1.4.5}$$

The transfer function of this filter is

$$H(\omega) = \sum_{k=-\infty}^{\infty} h_k e^{-i\omega k} \tag{1.4.6}$$

Throughout the text, we will follow the convention of writing $H(z)$ for the convolution operator of a linear system and its corresponding Z-transform and writing $H(\omega)$ for its transfer function. We obtain the transfer function $H(\omega)$ from $H(z)$ by the substitution $z = e^{i\omega}$:

$$\boxed{H(\omega) = H(z)\big|_{z=e^{i\omega}}}$$

We recognize the slight abuse of notation in writing $H(\omega)$ instead of $H(e^{i\omega})$ and in using z as both an operator and a complex variable, but we prefer the simplicity of notation it affords.

From (1.4.5) and (1.3.2), we obtain

$$\begin{aligned} r_y(k) &= \sum_{p=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h_p h_m^* E\left\{e(t-p)e^*(t-m-k)\right\} \\ &= \sum_{p=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h_p h_m^* r_e(m+k-p) \end{aligned} \tag{1.4.7}$$

Inserting (1.4.7) in (1.3.7) gives

$$\begin{aligned}\phi_y(\omega) &= \sum_{k=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h_p h_m^* r_e(m+k-p) e^{-i\omega(k+m-p)} e^{i\omega m} e^{-i\omega p} \\ &= \left[\sum_{p=-\infty}^{\infty} h_p e^{-i\omega p}\right] \left[\sum_{m=-\infty}^{\infty} h_m^* e^{i\omega m}\right] \left[\sum_{\tau=-\infty}^{\infty} r_e(\tau) e^{-i\omega\tau}\right] \\ &= |H(\omega)|^2 \phi_e(\omega) \qquad (1.4.8)\end{aligned}$$

From (1.4.8), we get the important formula

$$\boxed{\phi_y(\omega) = |H(\omega)|^2 \phi_e(\omega)} \qquad (1.4.9)$$

which will be much used in the next chapters.

Finally, we derive a property that will be of use in Chapter 5. Let the signals $y(t)$ and $x(t)$ be related by

$$y(t) = e^{i\omega_0 t} x(t) \qquad (1.4.10)$$

for some given ω_0. Then, it holds that

$$\boxed{\phi_y(\omega) = \phi_x(\omega - \omega_0)} \qquad (1.4.11)$$

In other words, multiplication of a temporal sequence by $e^{i\omega_0 t}$ shifts its spectral density by the angular frequency ω_0. This interpretation motivates calling the process of constructing $y(t)$, as in (1.4.10), *complex (de)modulation*. The proof of (1.4.11) is immediate: Equations (1.4.10) and (1.3.2) imply that

$$r_y(k) = e^{i\omega_0 k} r_x(k) \qquad (1.4.12)$$

so we obtain

$$\phi_y(\omega) = \sum_{k=-\infty}^{\infty} r_x(k) e^{-i(\omega-\omega_0)k} = \phi_x(\omega - \omega_0) \qquad (1.4.13)$$

which is the desired result.

1.5 THE SPECTRAL ESTIMATION PROBLEM

The spectral estimation problem can now be stated more formally as follows:

> From a finite-length record $\{y(1), \ldots, y(N)\}$ of a second-order stationary random process, find an estimate $\hat{\phi}(\omega)$ of its power spectral density $\phi(\omega)$, for $\omega \in [-\pi, \pi]$. (1.5.1)

It would, of course, be desirable that $\hat{\phi}(\omega)$ be as close to $\phi(\omega)$ as possible. As we shall see, the main limitation on the quality of most PSD estimates is due to the quite small number of data samples usually available for processing. Note that N will be used throughout this text to denote the number of points of the available data sequence. In some applications, N is small because the cost of obtaining large amounts of data is prohibitive. Most commonly, the value of N is limited by the fact that the signal under study can be considered second-order stationary only over short observation intervals.

As already mentioned in the introductory part of this chapter, there are two main approaches to the PSD estimation problem. The *nonparametric approach*, discussed in Chapters 2 and 5, proceeds to estimate the PSD by relying essentially only on the basic definitions (1.3.7) and (1.3.10) and on some properties that follow directly from these definitions. In particular, these methods do not impose any assumption on the functional form of $\phi(\omega)$. This is in contrast with the *parametric approach*, discussed in Chapters 3, 4, and 6. That approach makes assumptions on the signal under study, which leads to a parameterized functional form of the PSD, and then proceeds by estimating the parameters in the PSD model. The parametric approach can thus be used only when there is enough information about the studied signal to allow formulation of a model. Otherwise, the nonparametric approach should be used. Interestingly enough, the nonparametric methods are close competitors to the parametric ones, even when the model form assumed by the latter is a reasonable description of reality.

1.6 COMPLEMENTS

1.6.1 Coherence Spectrum

Let

$$\boxed{C_{yu}(\omega) = \frac{\phi_{yu}(\omega)}{[\phi_{yy}(\omega)\phi_{uu}(\omega)]^{1/2}}} \tag{1.6.1}$$

denote the so-called *complex coherence* of the stationary signals $y(t)$ and $u(t)$. In the previous definition, $\phi_{yu}(\omega)$ is the cross-spectrum of the two signals, defined as the DTFT of the cross-correlation sequence $r_{yu}(k) = E\{y(t)u^*(t-k)\}$, and $\phi_{yy}(\omega)$ and $\phi_{uu}(\omega)$ are their respective PSDs. (We implicitly assume in (1.6.1) that $\phi_{yy}(\omega)$ and $\phi_{uu}(\omega)$ are strictly positive for all ω.) Also, let

$$\epsilon(t) = y(t) - \sum_{k=-\infty}^{\infty} h_k u(t-k) \tag{1.6.2}$$

denote the residues of the least-squares problem in Exercise 1.11. Hence, $\{h_k\}$ in equation (1.6.2) satisfies

$$\sum_{k=-\infty}^{\infty} h_k e^{-i\omega k} \triangleq H(\omega) = \phi_{yu}(\omega)/\phi_{uu}(\omega).$$

In what follows, we will show that

$$E\left\{|\epsilon(t)|^2\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} (1 - |C_{yu}(\omega)|^2)\phi_{yy}(\omega)\, d\omega \tag{1.6.3}$$

where $|C_{yu}(\omega)|$ is the so-called *coherence spectrum*. We will deduce from (1.6.3) that the coherence spectrum shows the extent to which $y(t)$ and $u(t)$ are linearly related to one another, hence providing a motivation for the name given to $|C_{yu}(\omega)|$. We will also show that $|C_{yu}(\omega)| \leq 1$, with equality, for all ω values, if and only if $y(t)$ and $u(t)$ are related as in equation (1.6.2) with $\epsilon(t) \equiv 0$ (in the mean-square sense). Finally, we will show that $|C_{yu}(\omega)|$ is invariant to linear filtering of $u(t)$ and $y(t)$ (possibly by different filters); that is, if $\tilde{u} = g * u$ and $\tilde{y} = f * y$, where f and g are linear filters and $*$ denotes convolution, then $|C_{\tilde{y}\tilde{u}}(\omega)| = |C_{yu}(\omega)|$.

Let $x(t) = \sum_{k=-\infty}^{\infty} h_k u(t-k)$. It can be shown that $u(t-k)$ and $\epsilon(t)$ are uncorrelated with one another for all k. (The reader is required to verify this claim; see also Exercise 1.11). Hence, $x(t)$ and $\epsilon(t)$ are also uncorrelated with each other. Now,

$$y(t) = \epsilon(t) + x(t), \tag{1.6.4}$$

so it follows that

$$\phi_{yy}(\omega) = \phi_{\epsilon\epsilon}(\omega) + \phi_{xx}(\omega) \tag{1.6.5}$$

By using the fact that $\phi_{xx}(\omega) = |H(\omega)|^2\phi_{uu}(\omega)$, we can write

$$\begin{aligned}
E\left\{|\epsilon(t)|^2\right\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi_{\epsilon\epsilon}(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[1 - |H(\omega)|^2 \frac{\phi_{uu}(\omega)}{\phi_{yy}(\omega)}\right] \phi_{yy}(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[1 - \frac{|\phi_{yu}(\omega)|^2}{\phi_{uu}(\omega)\phi_{yy}(\omega)}\right] \phi_{yy}(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[1 - |C_{yu}(\omega)|^2\right] \phi_{yy}(\omega)\, d\omega
\end{aligned}$$

which is (1.6.3).

Since the left-hand side in (1.6.3) is nonnegative and the PSD function $\phi_{yy}(\omega)$ is arbitrary, we must have $|C_{yu}(\omega)| \leq 1$ for all ω. It can also be seen from (1.6.3) that the closer $|C_{yu}(\omega)|$ is to 1, the smaller is the residual variance. In particular, if $|C_{yu}(\omega)| \equiv 1$, then $\epsilon(t) \equiv 0$ (in the mean-square sense) and hence $y(t)$ and $u(t)$ must be linearly related, as in (1.7.11). The previous interpretation leads to calling $C_{yu}(\omega)$ *the correlation coefficient in the frequency domain.*

Next, consider the filtered signals

$$\tilde{y}(t) = \sum_{k=-\infty}^{\infty} f_k y(t-k)$$

and

$$\tilde{u}(t) = \sum_{k=-\infty}^{\infty} g_k u(t-k)$$

where the filters $\{f_k\}$ and $\{g_k\}$ are assumed to be stable. As

$$\begin{aligned} r_{\tilde{y}\tilde{u}}(p) &\triangleq E\left\{\tilde{y}(t)\tilde{u}^*(t-p)\right\} \\ &= \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_k g_j^* E\left\{y(t-k)u^*(t-j-p)\right\} \\ &= \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_k g_j^* r_{yu}(j+p-k), \end{aligned}$$

it follows that

$$\begin{aligned} \phi_{\tilde{y}\tilde{u}}(\omega) &= \sum_{p=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_k e^{-i\omega k} \; g_j^* e^{i\omega j} \; r_{yu}(j+p-k) e^{-i\omega(j+p-k)} \\ &= \left(\sum_{k=-\infty}^{\infty} f_k e^{-i\omega k}\right) \left(\sum_{j=-\infty}^{\infty} g_j e^{-i\omega j}\right)^* \left(\sum_{s=-\infty}^{\infty} r_{yu}(s) e^{-i\omega s}\right) \\ &= F(\omega) G^*(\omega) \phi_{yu}(\omega) \end{aligned}$$

Hence,

$$|C_{\tilde{y}\tilde{u}}(\omega)| = \frac{|F(\omega)| \, |G(\omega)| \, |\phi_{yu}(\omega)|}{|F(\omega)| \phi_{yy}^{1/2}(\omega) |G(\omega)| \phi_{uu}^{1/2}(\omega)} = |C_{yu}(\omega)|$$

which is the desired result. Observe that the latter result is similar to the invariance of the modulus of the correlation coefficient in the time domain,

$$\frac{|r_{yu}(k)|}{[r_{yy}(0) r_{uu}(0)]^{1/2}}$$

to a scaling of the two signals: $\tilde{y}(t) = f \cdot y(t)$ and $\tilde{u}(t) = g \cdot u(t)$.

1.7 EXERCISES

Exercise 1.1: Scaling of the Frequency Axis

In this text, the time variable t has been expressed in units of the sampling interval T_s (say). Consequently, the frequency is measured in cycles per sampling interval. Assume we want the frequency units to be expressed in radians per second or in Hertz (Hz = cycles per second). Then we have to introduce the scaled frequency variables

$$\bar{\omega} = \omega/T_s \quad \bar{\omega} \in [-\pi/T_s,\ \pi/T_s] \text{ rad/sec} \tag{1.7.1}$$

and $\bar{f} = \bar{\omega}/2\pi$ (in Hz). It might be thought that the PSD in the new frequency variable is obtained by inserting $\omega = \bar{\omega}T_s$ into $\phi(\omega)$, but this is *wrong*. Show that the PSD, *as expressed in units of power per Hz*, is in fact given by

$$\boxed{\bar{\phi}(\bar{\omega}) = T_s\phi(\bar{\omega}T_s) \triangleq T_s \sum_{k=-\infty}^{\infty} r(k)e^{-i\bar{\omega}T_s k}, \qquad |\bar{\omega}| \le \pi/T_s} \tag{1.7.2}$$

(See [Marple 1987] for more details on this scaling aspect.)

Exercise 1.2: Time–Frequency Distributions

Let $y(t)$ denote a discrete-time signal, and let $Y(\omega)$ be its discrete-time Fourier transform. As explained in Section 1.2, $Y(\omega)$ shows how the energy in the *whole sequence* $\{y(t)\}_{t=-\infty}^{\infty}$ is distributed over frequency.

Assume that we want to characterize how the energy of the signal is distributed in *time and frequency*. If $D(t, \omega)$ is a function that characterizes the time–frequency distribution, then it should satisfy the so-called *marginal properties*:

$$\sum_{t=-\infty}^{\infty} D(t, \omega) = |Y(\omega)|^2 \tag{1.7.3}$$

and

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} D(t, \omega)d\omega = |y(t)|^2 \tag{1.7.4}$$

Use intuitive arguments to explain why the previous conditions are desirable properties of a time–frequency distribution. Next, show that the so-called Rihaczek distribution,

$$\boxed{D(t, \omega) = y(t)Y^*(\omega)e^{-i\omega t}} \tag{1.7.5}$$

satisfies conditions (1.7.3) and (1.7.4). (For treatments of the time–frequency distributions, the reader is referred to [Therrien 1992] and [Cohen 1995].)

Exercise 1.3: Two Useful Z-Transform Properties

(a) Let h_k be an absolutely summable sequence, and let $H(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k}$ be its Z-transform. Find the Z-transforms of the following two sequences:
(i) h_{-k}
(ii) $g_k = \sum_{m=-\infty}^{\infty} h_m h_{m-k}^*$.

(b) Show that, if z_i is a zero of $A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$, then $(1/z_i^*)$ is a zero of $A^*(1/z^*)$ (where $A^*(1/z^*) = [A(1/z^*)]^*$).

Exercise 1.4: A Simple ACS Example

Let $y(t)$ be the output of a linear system, as in Figure 1.1, with filter $H(z) = (1 + b_1 z^{-1})/(1 + a_1 z^{-1})$ whose input is zero-mean white noise with variance σ^2. (The ACS of such an input is $\sigma^2 \delta_{k,0}$.)

(a) Find $r(k)$ and $\phi(\omega)$ analytically in terms of a_1, b_1, and σ^2.

(b) Verify that $r(-k) = r^*(k)$ and that $|r(k)| \le r(0)$. Also verify that, when a_1 and b_1 are real, $r(k)$ can be written as a function of $|k|$.

Exercise 1.5: Alternative Proof that $|r(k)| \le r(0)$

We stated in the text that (1.3.4) follows from (1.3.6). Provide a proof of that statement. Also, find an alternative, simple proof of (1.3.4) by using (1.3.8).

Exercise 1.6: A Double Summation Formula

A result often used in the study of discrete-time random signals is the summation formula

$$\boxed{\sum_{t=1}^{N} \sum_{s=1}^{N} f(t-s) = \sum_{\tau=-N+1}^{N-1} (N - |\tau|) f(\tau)} \tag{1.7.6}$$

where $f(\cdot)$ is an arbitrary function. Provide a proof of this formula.

Exercise 1.7: Is a Truncated Autocovariance Sequence (ACS) a Valid ACS?

Suppose that $\{r(k)\}_{k=-\infty}^{\infty}$ is a valid ACS; thus, $\sum_{k=-\infty}^{\infty} r(k) e^{-i\omega k} \ge 0$ for all ω. Is it possible that, for some integer p, the partial (or truncated) sum

$$\sum_{k=-p}^{p} r(k) e^{-i\omega k}$$

is negative for some ω? Justify your answer.

Exercise 1.8: When Is a Sequence an Autocovariance Sequence?

We showed in Section 1.3 that, if $\{r(k)\}_{k=-\infty}^{\infty}$ is an ACS, then $R_m \geq 0$ for $m = 0, 1, 2, \ldots$. We also implied that the first definition of the PSD in (1.3.7) satisfies $\phi(\omega) \geq 0$ for all ω; however, we did not prove this by using (1.3.7) solely. Show that

$$\phi(\omega) = \sum_{k=-\infty}^{\infty} r(k)e^{-i\omega k} \geq 0 \text{ for all } \omega$$

if and only if

$$a^* R_m a \geq 0 \quad \text{for every } m \text{ and for every vector } a$$

Exercise 1.9: Spectral Density of the Sum of Two Correlated Signals

Let $y(t)$ be the output to the system shown in Figure 1.2. Assume $H_1(z)$ and $H_2(z)$ are linear, asymptotically stable systems. The inputs $e_1(t)$ and $e_2(t)$ are each zero-mean white noise, with

$$E\left\{\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}\begin{bmatrix} e_1^*(s) & e_2^*(s) \end{bmatrix}\right\} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}\delta_{t,s}$$

(a) Find the PSD of $y(t)$.
(b) Show that, for $\rho = 0$, $\phi_y(\omega) = \phi_{x_1}(\omega) + \phi_{x_2}(\omega)$.
(c) Show that, for $\rho = \pm 1$ and $\sigma_1^2 = \sigma_2^2 = \sigma^2$, $\phi_y(\omega) = \sigma^2|H_1(\omega) \pm H_2(\omega)|^2$.

Exercise 1.10: Least-Squares Spectral Approximation

Assume we are given an ACS $\{r(k)\}_{k=-\infty}^{\infty}$ or, equivalently, a PSD function $\phi(\omega)$, as in equation (1.3.7). We wish to find a finite-impulse response (FIR) filter, as in Figure 1.1, where $H(\omega) = h_0 + h_1 e^{-i\omega} + \ldots + h_m e^{-im\omega}$, whose input $e(t)$ is zero-mean unit-variance white noise and such

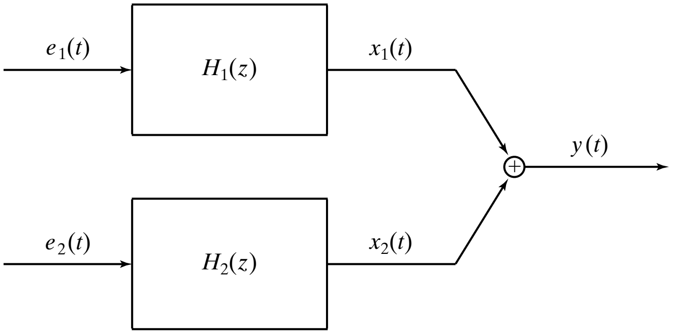


Figure 1.2 The system considered in Exercise 1.9.

that the output sequence $y(t)$ has PSD $\phi_y(\omega)$ "close to" $\phi(\omega)$. Specifically, we wish to find $h = [h_0 \dots h_m]^T$ so that the approximation error

$$\epsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\phi(\omega) - \phi_y(\omega)]^2 \, d\omega \tag{1.7.7}$$

is minimum.

(a) Show that ϵ is a quartic (fourth-order) function of h and that thus no simple closed-form solution h exists to minimize (1.7.7).

(b) We attempt to reparameterize the minimization problem as follows: We note that $r_y(k) \equiv 0$ for $|k| > m$; thus,

$$\phi_y(\omega) = \sum_{k=-m}^{m} r_y(k) e^{-i\omega k} \tag{1.7.8}$$

Equation (1.7.8), and the fact that $r_y(-k) = r_y^*(k)$, mean that $\phi_y(\omega)$ is a function of $g = [r_y(0) \dots r_y(m)]^T$. Show that the minimization problem in (1.7.7) is quadratic in g; it thus admits a closed-form solution. Show that the vector g that minimizes ϵ in equation (1.7.7) gives

$$r_y(k) = \begin{cases} r(k), & |k| \le m \\ 0, & \text{otherwise} \end{cases} \tag{1.7.9}$$

(c) Can you identify any problems with the "solution" (1.7.9)?

Exercise 1.11: Linear Filtering and the Cross-Spectrum

For two stationary signals $y(t)$ and $u(t)$, with cross-covariance sequence $r_{yu}(k) = E\{y(t) u^*(t-k)\}$, the *cross-spectrum* is defined as

$$\boxed{\phi_{yu}(\omega) = \sum_{k=-\infty}^{\infty} r_{yu}(k) e^{-i\omega k}} \tag{1.7.10}$$

Let $y(t)$ be the output of a linear filter with input $u(t)$,

$$y(t) = \sum_{k=-\infty}^{\infty} h_k u(t-k) \tag{1.7.11}$$

Show that the input PSD, $\phi_{uu}(\omega)$, the filter transfer function

$$H(\omega) = \sum_{k=-\infty}^{\infty} h_k e^{-i\omega k}$$

and $\phi_{yu}(\omega)$ are related through the so-called *Wiener–Hopf equation*:

$$\boxed{\phi_{yu}(\omega) = H(\omega)\phi_{uu}(\omega)} \tag{1.7.12}$$

Next, consider the least-squares (LS) problem

$$\min_{\{h_k\}} E\left\{\left|y(t) - \sum_{k=-\infty}^{\infty} h_k u(t-k)\right|^2\right\} \tag{1.7.13}$$

where now $y(t)$ and $u(t)$ are no longer necessarily related through equation (1.7.11). Show that the filter minimizing the preceding LS criterion is still given by the Wiener–Hopf equation, by minimizing the expectation in (1.7.13) with respect to the real and imaginary parts of h_k. (Assume that $\phi_{uu}(\omega) > 0$ for all ω.)

COMPUTER EXERCISES

Exercise C1.12: Computer Generation of Autocovariance Sequences

Autocovariance sequences are two-sided sequences. In this exercise, we develop computer techniques for generating two-sided ACSs.

Let $y(t)$ be the output of the linear system in Figure 1.1, with filter $H(z) = (1 + b_1 z^{-1})/(1 + a_1 z^{-1})$, whose input is zero-mean white noise with variance σ^2.

(a) Find $r(k)$ analytically in terms of a_1, b_1, and σ^2. (See also Exercise 1.4.)

(b) Plot $r(k)$ for $-20 \le k \le 20$ and for various values of a_1 and b_1. Notice that the tails of $r(k)$ decay at a rate dictated by $|a_1|$.

(c) When $a_1 \simeq b_1$ and $\sigma^2 = 1$, then $r(k) \simeq \delta_{k,0}$. Verify this for $a_1 = -0.95$, $b_1 = -0.9$, and for $a_1 = -0.75$, $b_1 = -0.7$.

(d) A quick way to generate (approximately) $r(k)$ on the computer is to use the fact that $r(k) = \sigma^2 h(k) * h^*(-k)$, where $h(k)$ is the impulse response of the filter in Figure 1.1 (see equation (1.4.7)) and $*$ denotes convolution. Consider the case where

$$H(z) = \frac{1 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}$$

Write a MATLAB function `genacs.m` whose inputs are M, σ^2, a, and b, where a and b are the vectors of denominator and numerator coefficients, respectively, and whose output is a vector of ACS coefficients from 0 to M. Your function should make use of the MATLAB functions `filter` (to generate $\{h_k\}_{k=0}^{M}$) and `conv` (to compute $r(k) = \sigma^2 h(k) * h^*(-k)$ by using the truncated impulse response sequence).

(e) Test your function, using $\sigma^2 = 1$, $a_1 = -0.9$, and $b_1 = 0.8$. Try $M = 20$ and $M = 150$; why is the result more accurate for larger M? Suggest a "rule of thumb" about a good choice of M in relation to the poles of the filter.

This method is a "quick and simple" way to compute an approximation to the ACS, but it is sometimes not very accurate because the impulse response is truncated. Methods for computing the exact ACS from σ^2, a, and b are discussed in Exercise 3.2 and also in [Kinkel, Perl, Scharf, and Stubberud 1979; Demeure and Mullis 1989].

Exercise C1.13: DTFT Computations Using Two-Sided Sequences

In this exercise, we consider the DTFT of two-sided sequences (including autocovariance sequences); in doing so, we illustrate some basic properties of autocovariance sequences.

(a) We first consider how to use the DTFT to determine $\phi(\omega)$ from $r(k)$ on a computer. We are given an ACS:

$$r(k) = \begin{cases} \dfrac{M - |k|}{M}, & |k| \le M \\ 0, & \text{otherwise} \end{cases} \tag{1.7.14}$$

Generate $r(k)$ for $M = 10$. Form, in MATLAB, a vector `x` of length $L = 256$ as

$$\texttt{x} = [r(0), r(1), \ldots, r(M), 0 \ldots, 0, r(-M), \ldots, r(-1)]$$

Verify that `xf=fft(x)` gives $\phi(\omega_k)$ for $\omega_k = 2\pi k/L$. (Note that the elements of `xf` should be nonnegative and real.) Explain why this particular choice of `x` is needed, citing appropriate circular shift and zero-padding properties of the DTFT.

Note that `xf` often contains a very small imaginary part due to computer roundoff error; replacing `xf` by `real(xf)` truncates this imaginary component and leads to more expected results when plotting.

A word of caution—do not truncate the imaginary part unless you are sure it is negligible; the command `zf=real(fft(z))` when

$$\texttt{z} = [r(-M), \ldots, r(-1), r(0), r(1), \ldots, r(M), 0 \ldots, 0]$$

gives erroneous "spectral" values; try it and explain why it does not work.

(b) Alternatively, since we can readily derive the analytical expression for $\phi(\omega)$, we can instead work backwards. Form a vector

$$\texttt{yf} = [\phi(0), \phi(2\pi/L), \phi(4\pi/L), \ldots, \phi((L-1)2\pi/L)]$$

and find `y=ifft(yf)`. Verify that `y` closely approximates the ACS.

(c) Consider the ACS $r(k)$ in Exercise C1.12; let $a_1 = -0.9$ and $b_1 = 0$, and set $\sigma^2 = 1$. Form a vector `x` as before, with $M = 10$, and find `xf`. Why is `xf` not a good approximation of $\phi(\omega_k)$ in this case? Repeat the experiment for $M = 127$ and $L = 256$; is the approximation better for this case? Why?

We can again work backwards from the analytical expression for $\phi(\omega)$. Form a vector

$$\texttt{yf} = [\phi(0), \phi(2\pi/L), \phi(4\pi/L), \ldots, \phi((L-1)2\pi/L)]$$

and find `y=ifft(yf)`. Verify that `y` closely approximates the ACS for large L (say, $L = 256$), but poorly approximates the ACS for small L (say, $L = 20$). By citing properties of inverse DTFTs of infinite, two-sided sequences, explain how the elements of `y` relate to the ACS $r(k)$ and why the approximation is poor for small L. Based on this explanation, give a "rule of thumb" for a choice of L that gives a good approximations of the ACS.

(d) We have seen that the `fft` command results in spectral estimates for $\omega \in [0, 2\pi)$ instead of the more commonly-used range $\omega \in [-\pi, \pi)$. The MATLAB command `fftshift` can be used to exchange the first and second halves of the `fft` output to make it correspond to the frequency range from $\omega \in [-\pi, \pi)$. Similarly, `fftshift` can be used on the output of the `ifft` operation to "center" the zero lag of an ACS. Experiment with `fftshift` to achieve both of these results. What frequency vector `w` is needed so that the command `plot(w, fftshift(fft(x)))` gives the spectral values at the proper frequencies? Similarly, what time vector `t` is needed to get a proper plot of the ACS with `stem(t,fftshift(ifft(xf)))`? Do the results depend on whether the vectors are even or odd in length?

Exercise C1.14: Relationship between the PSD and the Eigenvalues of the ACS Matrix

An interesting property of the ACS matrix R in equation (1.3.5) is that, for large dimensions m, its eigenvalues are close to the values of the PSD $\phi(\omega_k)$ for $\omega_k = 2\pi k/m$, $k = 0, 1, \ldots, m-1$. (See, for example, [Gray 1972].) We verify this property here:

Consider the ACS in Exercise C1.12, with the values $a_1 = -0.9$, $b_1 = 0.8$, and $\sigma^2 = 1$.

(a) Compute a vector `phi` that contains the values of $\phi(\omega_k)$ for $\omega_k = 2\pi k/m$, with $m = 256$ and $k = 0, 1, \ldots, m-1$. Plot a histogram of these values with `hist(phi)`. Also useful is the cumulative distribution of the values of `phi` (plotted on a logarithmic scale), which can be found with the command `semilogy( (1/m:1/m:1), sort(phi) )`.

(b) Compute the eigenvalues of R in equation (1.3.5) for various values of m. Plot the histogram of the eigenvalues and plot their cumulative distribution. Verify that, as m increases, the cumulative distribution of the eigenvalues approaches the cumulative distribution of the $\phi(\omega)$ values. Similarly, the histograms also approach the histogram for $\phi(\omega)$, but it is easier to see this result by using cumulative distributions than by using histograms.

2

Nonparametric Methods

2.1 INTRODUCTION

The nonparametric methods of spectral estimation rely entirely on the definitions (1.3.7) and (1.3.10) of PSD to provide spectral estimates. These methods constitute the “classical means” for PSD estimation [JENKINS AND WATTS 1968]. The present chapter reviews the main nonparametric methods, their properties, and the Fast Fourier Transform (FFT) algorithm for their implementation. A related discussion is to be found in Chapter 5, where the nonparametric approach to PSD estimation is given a filter-bank interpretation.

We first introduce two common spectral estimators, the *periodogram* and the *correlogram*, derived directly from (1.3.10) and (1.3.7), respectively. These methods are then shown to be equivalent under weak conditions. The periodogram and correlogram methods provide reasonably high resolution for sufficiently long data lengths, but are poor spectral estimators, because their variance is high and does not decrease with increasing data length. (In Chapter 5, we provide an interpretation of the periodogram and correlogram methods as a power estimate based on a *single* sample of a filtered version of the signal under study; it is thus not surprising that the periodogram or correlogram variance is large.)

The high variance of the periodogram and correlogram methods motivates the development of modified methods that have lower variance, at the cost of reduced resolution. Several modified methods have been introduced, and we present some of the most popular ones. We show them all to be, more or less, equivalent in their properties and performance for large data lengths.

2.2 PERIODOGRAM AND CORRELOGRAM METHODS

2.2.1 Periodogram

The periodogram method relies on the definition (1.3.10) of the PSD. Neglecting the expectation and the limit operation in (1.3.10), which cannot be performed when the only available information on the signal consists of the samples $\{y(t)\}_{t=1}^{N}$, we get

$$\boxed{\hat{\phi}_p(\omega) = \frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\omega t}\right|^2 \qquad \text{(Periodogram)}} \tag{2.2.1}$$

One of the first uses of the *periodogram spectral estimator*, (2.2.1), has been in determining possible "hidden periodicities" in time series, which may be seen as a motivation for the name of this method [SCHUSTER 1900].

2.2.2 Correlogram

The correlation-based definition (1.3.7) of the PSD leads to the *correlogram spectral estimator* [BLACKMAN AND TUKEY 1959]:

$$\boxed{\hat{\phi}_c(\omega) = \sum_{k=-(N-1)}^{N-1} \hat{r}(k)e^{-i\omega k} \qquad \text{(Correlogram)}} \tag{2.2.2}$$

where $\hat{r}(k)$ denotes an estimate of the covariance lag $r(k)$, obtained from the available sample $\{y(1), \ldots, y(N)\}$. When no assumption is made on the signal under study, except for the stationarity assumption, there are two standard ways to obtain the sample covariances required in (2.2.2):

$$\hat{r}(k) = \frac{1}{N-k}\sum_{t=k+1}^{N} y(t)y^*(t-k), \qquad 0 \le k \le N-1 \tag{2.2.3}$$

and

$$\boxed{\hat{r}(k) = \frac{1}{N}\sum_{t=k+1}^{N} y(t)y^*(t-k) \qquad 0 \le k \le N-1} \tag{2.2.4}$$

The sample covariances for negative lags are then constructed by using the property (1.3.3) of the covariance function:

$$\hat{r}(-k) = \hat{r}^*(k), \qquad k = 0, \ldots, N-1 \tag{2.2.5}$$

The estimator (2.2.3) is called the standard unbiased ACS estimate, and (2.2.4) is called the standard biased ACS estimate. The biased ACS estimate is most commonly used, for the following reasons:

- For most stationary signals, the covariance function decays rather rapidly, so that $r(k)$ is quite small for large lags k. Comparing the definitions (2.2.3) and (2.2.4), it can be seen that $\hat{r}(k)$ in (2.2.4) will be small for large k (provided N is reasonably large), whereas $\hat{r}(k)$ in (2.2.3) could take large and erratic values for large k, as it is obtained by averaging only a few products in such a case (in particular, only one product for $k = N - 1$!). This observation implies that (2.2.4) is likely to be a more accurate estimator of $r(k)$ than (2.2.3) for medium and large values of k (compared with N). For small values of k, the two estimators in (2.2.3) and (2.2.4) can be expected to behave in a similar manner.
- The sequence $\{\hat{r}(k), k = 0, \pm 1, \pm 2, \ldots\}$ obtained with (2.2.4) is guaranteed to be positive semidefinite (as it should, see equation (1.3.5) and the related discussion), but this is not the case for (2.2.3). This fact is especially important for PSD estimation, since a sample covariance sequence that is not positive definite, when inserted in (2.2.2), may lead to negative spectral estimates, and this is undesirable in most applications.

When the sample covariances (2.2.4) are inserted in (2.2.2), it can be shown that the spectral estimate so obtained is identical to (2.2.1). In other words, we have the following result:

$$\boxed{\begin{array}{l}\hat{\phi}_c(\omega)\text{, when evaluated by using the standard biased ACS estimates,}\\ \text{coincides with } \hat{\phi}_p(\omega).\end{array}} \qquad (2.2.6)$$

A simple proof of (2.2.6) runs as follows: Consider the signal

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} y(k) e(t-k) \qquad (2.2.7)$$

where $\{y(k)\}$ are considered to be fixed (nonrandom) constants and $e(t)$ is a white noise of unit variance: $E\{e(t)e^*(s)\} = \delta_{t,s}$ ($= 1$ if $t = s$; and $= 0$ otherwise). Hence, $x(t)$ is the output of a filter with the following transfer function:

$$Y(\omega) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} y(k) e^{-i\omega k}$$

Since the PSD of the input to the filter is given by $\phi_e(\omega) = 1$, it follows from (1.4.5) that

$$\phi_x(\omega) = |Y(\omega)|^2 = \hat{\phi}_p(\omega) \qquad (2.2.8)$$

On the other hand, a straightforward calculation gives (for $k \geq 0$)

$$\begin{aligned} r_x(k) &= E\left\{x(t)x^*(t-k)\right\} \\ &= \frac{1}{N}\sum_{p=1}^{N}\sum_{s=1}^{N} y(p)y^*(s)E\left\{e(t-p)e^*(t-k-s)\right\} \\ &= \frac{1}{N}\sum_{p=1}^{N}\sum_{s=1}^{N} y(p)y^*(s)\delta_{p,k+s} = \frac{1}{N}\sum_{p=k+1}^{N} y(p)y^*(p-k) \\ &= \begin{cases} \hat{r}(k) \text{ given by (2.2.4)}, & k = 0, \ldots, N-1 \\ 0, & k \geq N \end{cases} \end{aligned} \tag{2.2.9}$$

Inserting (2.2.9) in the definition (1.3.7) of PSD yields the following alternative expression for $\phi_x(\omega)$:

$$\phi_x(\omega) = \sum_{k=-(N-1)}^{N-1} \hat{r}(k)e^{-i\omega k} = \hat{\phi}_c(\omega) \tag{2.2.10}$$

Comparing (2.2.8) and (2.2.10) concludes the proof of the claim (2.2.6).

The equivalence of the periodogram and correlogram spectral estimators can be used to derive their properties simultaneously. These two methods are shown in Section 2.4 to provide *poor estimates* of the PSD. There are two reasons for this, and both can be explained intuitively by using $\hat{\phi}_c(\omega)$:

- The estimation errors in $\hat{r}(k)$ are on the order of $1/\sqrt{N}$ for large N (see Exercise 2.4), at least for $|k|$ not too close to N. Because $\hat{\phi}_c(\omega) = \hat{\phi}_p(\omega)$ is a sum that involves $(2N-1)$ such covariance estimates, the difference between the true and estimated spectra will be a sum of "many small" errors. Hence, there is no guarantee that the total error will die out as N increases. The spectrum estimation error is even larger than what is suggested by the preceding discussion, because errors in $\{\hat{r}(k)\}$, for $|k|$ close to N, are typically of an order larger than $1/\sqrt{N}$. The consequence is that the variance of $\hat{\phi}_c(\omega)$ does not go to zero as N increases.
- In addition, if $r(k)$ converges slowly to zero, then the periodogram estimates will be biased. Indeed, for lags $|k| \simeq N$, $\hat{r}(k)$ will be a poor estimate of $r(k)$ since $\hat{r}(k)$ is the sum of only a few lag products that are divided by N (see equation (2.2.4)). Thus, $\hat{r}(k)$ will be much closer to zero than $r(k)$ is; in fact, $E\left\{\hat{r}(k)\right\} = [(N-|k|)/N]r(k)$, and the bias is significant for $|k| \simeq N$ if $r(k)$ is not close to zero in this region. If $r(k)$ decays rapidly to zero, the bias will be small and will not contribute significantly to the total error in $\hat{\phi}_c(\omega)$; however, the nonzero variance just discussed will still be present.

Both the bias and the variance of the periodogram are discussed more quantitatively in Section 2.4.

Another intuitive explanation for the poor statistical accuracy of the periodogram and correlogram methods is given in Chapter 5, where it is shown, roughly speaking, that these methods can be viewed as procedures attempting to estimate the variance of a data sequence from a *single* sample.

In spite of their poor quality as spectral estimators, the periodogram and correlogram methods form the basis for the improved nonparametric spectral estimation methods, to be discussed later in this chapter. As such, computation of these two basic estimators is relevant to many other nonparametric estimators derived from them. The next section addresses this computational task.

2.3 PERIODOGRAM COMPUTATION VIA FFT

In practice, it is not possible to evaluate $\hat{\phi}_p(\omega)$ (or $\hat{\phi}_c(\omega)$) over a continuum of frequencies. Hence, the frequency variable must be sampled for the purpose of computing $\hat{\phi}_p(\omega)$. The following frequency sampling scheme is most commonly used:

$$\boxed{\omega = \frac{2\pi}{N}k, \qquad k = 0, \ldots, N-1} \tag{2.3.1}$$

Define

$$W = e^{-i\frac{2\pi}{N}} \tag{2.3.2}$$

Then, evaluation of $\hat{\phi}_p(\omega)$ (or $\hat{\phi}_c(\omega)$) at the frequency samples in (2.3.1) basically reduces to the computation of the following Discrete Fourier Transform (DFT):

$$Y(k) = \sum_{t=1}^{N} y(t)W^{(t-1)k}, \qquad k = 0, \ldots, N-1 \tag{2.3.3}$$

A direct evaluation of (2.3.3) would require about N^2 complex multiplications and additions, which might be a prohibitive burden for large values of N. Any procedure that computes (2.3.3) in less than N^2 flops (1 flop = 1 complex multiplication plus 1 complex addition) is called a Fast Fourier Transform (FFT) algorithm. In recent years, there has been significant interest in developing more and more computationally efficient FFT algorithms. In the following, we review one of the first FFT procedures—the so-called radix-2 FFT—which, while not being the most computationally efficient of all, is easy to program in a computer and yet quite computationally efficient [Cooley and Tukey 1965; Proakis, Rader, Ling, and Nikias 1992].

2.3.1 Radix-2 FFT

Assume that N is a power of 2:

$$\boxed{N = 2^m} \tag{2.3.4}$$

If this is not the case, then we can resort to *zero padding*, as described in the next subsection. By splitting the sum in (2.3.3) into two parts, we get

$$\begin{aligned} Y(k) &= \sum_{t=1}^{N/2} y(t)W^{(t-1)k} + \sum_{t=N/2+1}^{N} y(t)W^{(t-1)k} \\ &= \sum_{t=1}^{N/2} [y(t) + y(t+N/2)W^{\frac{Nk}{2}}]W^{(t-1)k} \end{aligned} \tag{2.3.5}$$

Next, note that

$$W^{\frac{Nk}{2}} = \begin{cases} 1, & \text{for even } k \\ -1, & \text{for odd } k \end{cases} \tag{2.3.6}$$

Using this simple observation in (2.3.5), we obtain

$$\boxed{\begin{array}{l} \text{For } k = 2p = 0, 2, \ldots \\ \qquad\qquad Y(2p) = \displaystyle\sum_{t=1}^{\bar{N}} [y(t) + y(t+\bar{N})]\bar{W}^{(t-1)p} \end{array}} \tag{2.3.7}$$

$$\boxed{\begin{array}{l} \text{For } k = 2p+1 = 1, 3, \ldots \\ \qquad\qquad Y(2p+1) = \displaystyle\sum_{t=1}^{\bar{N}} \{[y(t) - y(t+\bar{N})]W^{t}\}\bar{W}^{(t-1)p} \end{array}} \tag{2.3.8}$$

where $\bar{N} = N/2$ and $\bar{W} = W^2 = e^{-i2\pi/\bar{N}}$.

These two equations are the core of the radix-2 FFT algorithm. Both of these equations represent DFTs for sequences of length equal to $\bar{N}$. Computation of the sequences transformed in (2.3.7) and (2.3.8) requires roughly $\bar{N}$ flops. Hence, the computation of an N-point transform has been reduced to the evaluation of two $N/2$-point transforms plus a sequence computation requiring about $N/2$ flops. This reduction process is continued until $\bar{N} = 1$ (which is made possible by requiring N to be a power of 2).

In order to evaluate the number of flops required by a radix-2 FFT, let c_k denote the computational cost (expressed in flops) of a 2^k-point radix-2 FFT. According to the discussion in the previous paragraph, c_k satisfies the recursion

$$c_k = 2^k/2 + 2c_{k-1} = 2^{k-1} + 2c_{k-1} \tag{2.3.9}$$

with initial condition $c_1 = 1$ (the number of flops required by a 1-point transform). By iterating (2.3.9), we obtain the solution

$$c_k = k2^{k-1} = \frac{1}{2}k2^k \tag{2.3.10}$$

from which it follows that $c_m = \frac{1}{2}m2^m = \frac{1}{2}N\log_2 N$; thus

$$\boxed{\text{An } N\text{-point radix-2 FFT requires about } \tfrac{1}{2}N\log_2 N \text{ flops.}} \tag{2.3.11}$$

As a comparison, the number of complex operations required to carry out an N-point *split-radix* FFT, which at present appears to be the most practical algorithm for general-purpose computers when N is a power of 2, is about $\frac{1}{3}N\log_2 N$. (See [Proakis, Rader, Ling, and Nikias 1992].)

2.3.2 Zero Padding

In some applications, N is not a power of 2 and so the previously described radix-2 FFT algorithm cannot be applied directly to the original data sequence. However, this is easily remedied; we may increase the length of the given sequence by means of zero padding $\{y(1), \ldots, y(N), 0, 0, \ldots\}$ until the length of the sequence so obtained is, say, L (which is generally chosen as a power of 2).

Zero padding is also useful when the frequency sampling (2.3.1) is considered to be too sparse to provide a good representation of the continuous-frequency estimated spectrum, for example $\hat{\phi}_p(\omega)$. Applying the FFT algorithm to the data sequence padded with zeroes gives

$$\hat{\phi}_p(\omega) \text{ at frequencies } \omega_k = \frac{2\pi k}{L}, \qquad 0 \le k \le L-1$$

The finer sampling of $\hat{\phi}(\omega)$ can reveal details in the spectrum that were not visible without zero padding.

Since the *continuous-frequency* spectral estimate, $\hat{\phi}_p(\omega)$, is the same for both the original data sequence and the sequence padded with zeroes, zero padding cannot of course improve the spectral resolution of the periodogram methods. See [Oppenheim and Schafer 1989; Porat 1997] for further discussion.

In a zero-padded data sequence, the number of nonzero data points may be considerably smaller than the total number of samples—that is, $N \ll L$. In such a case, a significant time saving can be obtained by *pruning the FFT* algorithm by reducing or eliminating operations on zeroes. (See, for example, [Markel 1971].) FFT pruning, along with a decimation in time, can also be used to reduce the computation time when we want to evaluate the FFT only in a narrow region of the frequency domain. (See [Markel 1971].)

2.4 PROPERTIES OF THE PERIODOGRAM METHOD

The analysis of the statistical properties of $\hat{\phi}_p(\omega)$ (or $\hat{\phi}_c(\omega)$) is important, in that it shows the poor quality of the periodogram as an estimator of the PSD and, in addition, provides some insight into how we can modify the periodogram so as to obtain better spectral estimators. We split the analysis in two parts: bias analysis and variance analysis. (See also [Priestley 1981].)

The bias and variance of an estimator are two measures often used to characterize its performance. A primary motivation is that the total squared error of the estimate is the sum of the squared bias and the variance. To see this, let a denote any quantity to be estimated, and let $\hat{a}$ be an estimate of a. Then the mean squared error (MSE) of the estimate is

$$\begin{aligned}\text{MSE} \triangleq E\left\{|\hat{a}-a|^2\right\} &= E\left\{\left|\hat{a}-E\left\{\hat{a}\right\}+E\left\{\hat{a}\right\}-a\right|^2\right\} \\ &= E\left\{\left|\hat{a}-E\left\{\hat{a}\right\}\right|^2\right\} + \left|E\left\{\hat{a}\right\}-a\right|^2 \\ &\qquad +2\,\text{Re}\left[\left(E\left\{\hat{a}-E\left\{\hat{a}\right\}\right\}\right)^* \left(E\left\{\hat{a}\right\}-a\right)\right] \\ &= \text{var}\{\hat{a}\} + \left|\text{bias}\{\hat{a}\}\right|^2 \end{aligned} \tag{2.4.1}$$

By considering the bias and variance components of the MSE separately, we gain some additional insight into the source of error and ways to reduce the error.

2.4.1 Bias Analysis of the Periodogram

By using the result (2.2.6), we obtain

$$E\left\{\hat{\phi}_p(\omega)\right\} = E\left\{\hat{\phi}_c(\omega)\right\} = \sum_{k=-(N-1)}^{N-1} E\left\{\hat{r}(k)\right\} e^{-i\omega k} \tag{2.4.2}$$

For $\hat{r}(k)$ defined in (2.2.4),

$$E\left\{\hat{r}(k)\right\} = \left(1-\frac{k}{N}\right) r(k), \qquad k \geq 0 \tag{2.4.3}$$

and

$$E\left\{\hat{r}(-k)\right\} = E\left\{\hat{r}^*(k)\right\} = \left(1-\frac{k}{N}\right) r(-k), \qquad -k \leq 0 \tag{2.4.4}$$

Hence,

$$E\left\{\hat{\phi}_p(\omega)\right\} = \sum_{k=-(N-1)}^{N-1} \left(1-\frac{|k|}{N}\right) r(k) e^{-i\omega k} \tag{2.4.5}$$

Define

$$w_B(k) = \begin{cases} 1 - \frac{|k|}{N}, & k = 0, \pm 1, \dots, \pm(N-1) \\ 0, & \text{otherwise} \end{cases} \tag{2.4.6}$$

The preceding sequence is called the *triangular window* or the *Bartlett window*. By using $w_B(k)$, we can write (2.4.5) as a DTFT:

$$E\left\{\hat{\phi}_p(\omega)\right\} = \sum_{k=-\infty}^{\infty} [w_B(k) r(k)] e^{-i\omega k} \tag{2.4.7}$$

The DTFT of the product of two sequences is equal to the convolution of their respective DTFTs. Hence, (2.4.7) leads to

$$\boxed{E\left\{\hat{\phi}_p(\omega)\right\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\psi) W_B(\omega - \psi) d\psi} \tag{2.4.8}$$

where $W_B(\omega)$ is the DTFT of the triangular window. For completeness, we include a direct proof of (2.4.8). Inserting (1.3.8) in (2.4.7), we get

$$E\left\{\hat{\phi}_p(\omega)\right\} = \sum_{k=-\infty}^{\infty} w_B(k) \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\psi) e^{i\psi k} d\psi\right] e^{-i\omega k} \tag{2.4.9}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\psi) \left[\sum_{k=-\infty}^{\infty} w_B(k) e^{-ik(\omega-\psi)}\right] d\psi \tag{2.4.10}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\psi) W_B(\omega - \psi) d\psi \tag{2.4.11}$$

which is (2.4.8).

We can find an explicit expression for $W_B(\omega)$ as follows:

$$W_B(\omega) = \sum_{k=-(N-1)}^{N-1} \frac{N - |k|}{N} e^{-i\omega k} \tag{2.4.12}$$

$$= \frac{1}{N} \sum_{t=1}^{N} \sum_{s=1}^{N} e^{-i\omega(t-s)} = \frac{1}{N} \left|\sum_{t=1}^{N} e^{i\omega t}\right|^2 \tag{2.4.13}$$

$$= \frac{1}{N} \left|\frac{e^{i\omega N} - 1}{e^{i\omega} - 1}\right|^2 = \frac{1}{N} \left|\frac{e^{i\omega N/2} - e^{-i\omega N/2}}{e^{i\omega/2} - e^{-i\omega/2}}\right|^2 \tag{2.4.14}$$

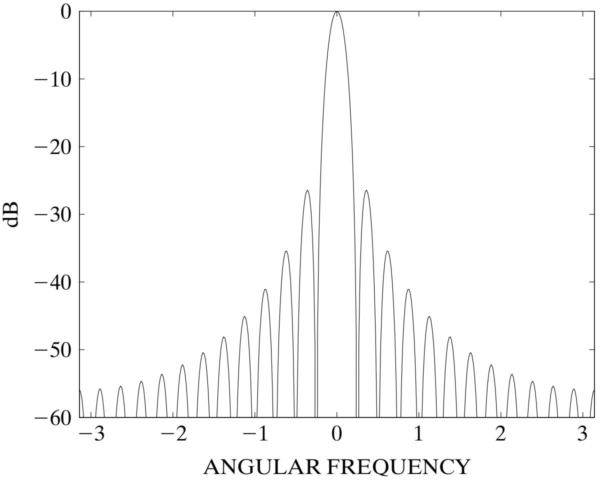


Figure 2.1 $W_B(\omega)/W_B(0)$, for $N = 25$.

or, in final form,

$$W_B(\omega) = \frac{1}{N}\left[\frac{\sin(\omega N/2)}{\sin(\omega/2)}\right]^2 \tag{2.4.15}$$

$W_B(\omega)$ is sometimes referred to as the *Fejer kernel.* As an illustration, $W_B(\omega)$ is displayed as a function of ω, in Figure 2.1.

The convolution formula (2.4.8) is the key equation for understanding the behavior of the mean estimated spectrum $E\{\hat{\phi}_p(\omega)\}$. In order to facilitate the interpretation of this equation, the reader may think of it as representing a dynamical system with “input” $\phi(\omega)$, “weighting function” $W_B(\omega)$, and “output” $E\{\hat{\phi}_p(\omega)\}$. Note that a similar equation would be obtained if the covariance estimator (2.2.3) were used in $\hat{\phi}_c(\omega)$, in lieu of (2.2.4). In that case, $E\{\hat{r}(k)\} = r(k)$, so the corresponding $W(\omega)$ function that would appear in (2.4.8) is the DTFT of the *rectangular window*

$$w_R(k) = \begin{cases} 1, & k = 0, \pm 1, \ldots, \pm(N-1) \\ 0, & \text{otherwise} \end{cases} \tag{2.4.16}$$

A straightforward calculation gives

$$\begin{aligned} W_R(\omega) &= \sum_{k=-(N-1)}^{(N-1)} e^{-i\omega k} = 2\,\mathrm{Re}\left[\frac{e^{iN\omega}-1}{e^{i\omega}-1}\right] - 1 \\ &= \frac{2\cos\left[\frac{(N-1)\omega}{2}\right]\sin\left[\frac{N\omega}{2}\right]}{\sin\left[\frac{\omega}{2}\right]} - 1 = \frac{\sin\left[\left(N-\frac{1}{2}\right)\omega\right]}{\sin\left[\frac{\omega}{2}\right]} \end{aligned} \tag{2.4.17}$$

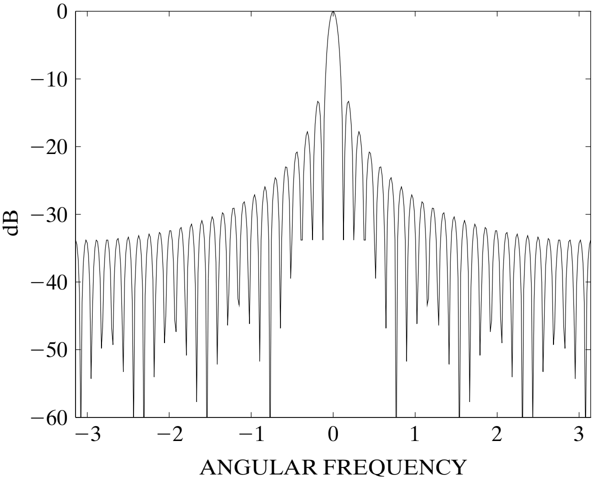


Figure 2.2 $W_R(\omega)/W_R(0)$, for $N = 25$.

which is displayed in Figure 2.2 (for $N = 25$; to facilitate comparison with $W_B(\omega)$). $W_R(\omega)$ is sometimes called the *Dirichlet kernel.*

As can be seen, there are no "essential" differences between $W_R(\omega)$ and $W_B(\omega)$. For conciseness, in the following we focus on the use of the triangular window.

Since we would like $E\{\hat{\phi}_p(\omega)\}$ to be as close to $\phi(\omega)$ as possible, it follows from (2.4.8) that $W_B(\omega)$ should be a close approximation to a Dirac impulse. The *half-power (3 dB) width* of the *main lobe* of $W_B(\omega)$ can be shown to be approximately $2\pi/N$ radians (see Exercise 2.15), so, in frequency units (with $f = \omega/2\pi$),

$$\boxed{\text{main lobe width in frequency } f \simeq 1/N} \tag{2.4.18}$$

(Also, see the calculation of the time-bandwidth product for windows in the next section, which supports (2.4.18).) It follows from (2.4.18) that $W_B(\omega)$ is a poor approximation of a Dirac impulse for small values of N. In addition, unlike the Dirac delta function, $W_B(\omega)$ has a large number of *sidelobes*. It follows that the bias of the periodogram spectral estimate can basically be divided into two components. These two components correspond respectively to the nonzero main lobe width and the nonzero sidelobe height of the window function $W_B(\omega)$, as we explain next.

The principal effect of the *main lobe* of $W_B(\omega)$ is to smear or smooth the estimated spectrum. Assume, for instance, that $\phi(\omega)$ has two peaks separated in frequency f by less than $1/N$. Then these two peaks appear as a single broader peak in $\mathrm{E}\{\hat{\phi}_p(\omega)\}$, because (per (2.4.8)), the "response" of the "system" corresponding to $W_B(\omega)$ to the first peak does not die out before the "response" to the second peak starts. This kind of effect of the main lobe on the estimated spectrum is called *smearing*. Smearing prevents the periodogram-based methods from resolving details in the studied spectrum that are separated in frequency by less than $1/N$ cycles per sampling interval. For this reason, $1/N$ is called the *spectral resolution limit* of the periodogram method.

Remark: The previous comments on resolution give us the occasion to stress that, in spite of the fact that we have seen the PSD as a function of the angular frequency (ω), we generally refer to *the resolution in frequency* (f) in units of cycles per sampling interval. Of course, the "resolution in angular frequency" is determined from the "resolution in frequency" by the simple relation $\omega = 2\pi f$. ■

The principal effect of the *sidelobes* on the estimated spectrum consists of transferring power from the frequency bands that concentrate most of the power in the signal to bands that contain less or no power. This effect is called *leakage*. For instance, a dominant peak in $\phi(\omega)$ could, through convolution with the sidelobes of $W_B(\omega)$, lead to an estimated spectrum that contains power in frequency bands where $\phi(\omega)$ is zero. Note that the smearing effect associated with the main lobe can also be interpreted as a form of leakage from a local peak of $\phi(\omega)$ to neighboring frequency bands.

It follows from the previous discussion that smearing and leakage are particularly critical for spectra with large amplitude ranges, such as peaky spectra. For smooth spectra, these effects are less important. In particular, we see from (2.4.7) that, for *white noise* (which has a maximally smooth spectrum), the periodogram is an *unbiased* spectral estimator: $E\{\hat{\phi}_p(\omega)\} = \phi(\omega)$. (See also Exercise 2.9.)

The bias of the periodogram estimator, even though it might be severe for spectra with large dynamic ranges when the sample length is small, does not constitute the main limitation of this spectral estimator. In fact, if the bias were the only problem, then increasing N (assuming this is possible) would cause the bias in $\hat{\phi}_p(\omega)$ to be eliminated. In order to see this, note from (2.4.5) that

$$\lim_{N\to\infty} E\left\{\hat{\phi}_p(\omega)\right\} = \phi(\omega)$$

Hence, the periodogram is an *asymptotically unbiased spectral estimator*. The main problem of the periodogram method lies in its large variance, as is explained next.

2.4.2 Variance Analysis of the Periodogram

The finite-sample variance of $\hat{\phi}_p(\omega)$ can be easily established only in some specific cases, such as in the case of Gaussian white noise. The *asymptotic* variance of $\hat{\phi}_p(\omega)$, however, can be derived for more general signals. In the following, we present an *asymptotic (for $N \gg 1$) analysis* of the variance of $\hat{\phi}_p(\omega)$, since it turns out to be sufficient for showing the poor statistical accuracy of the periodogram. (For a finite-sample analysis, see Exercise 2.13.)

Some preliminary discussion is required. A sequence $\{e(t)\}$ is called *complex (or circular) white noise* if it satisfies

$$\boxed{\begin{aligned} E\{e(t)e^*(s)\} &= \sigma^2\delta_{t,s} \\ E\{e(t)e(s)\} &= 0, \qquad \text{for all } t \text{ and } s \end{aligned}} \tag{2.4.19}$$

Note that $\sigma^2 = E\left\{|e(t)|^2\right\}$ is the variance (or power) of $e(t)$. Equation (2.4.19) can be rewritten as

$$\begin{cases} E\{\mathrm{Re}[e(t)]\,\mathrm{Re}[e(s)]\} &= \frac{\sigma^2}{2}\delta_{t,s} \\ E\{\mathrm{Im}[e(t)]\,\mathrm{Im}[e(s)]\} &= \frac{\sigma^2}{2}\delta_{t,s} \\ E\{\mathrm{Re}[e(t)]\,\mathrm{Im}[e(s)]\} &= 0 \end{cases} \tag{2.4.20}$$

Hence, the real and imaginary parts of a complex/circular white noise are real-valued white-noise sequences, of identical power equal to $\sigma^2/2$ and uncorrelated with one another. See Appendix B for more details on circular random sequences.

In what follows, we shall also make use of the symbol $\mathcal{O}(1/N^\alpha)$, for some $\alpha > 0$, to denote a random variable that is such that the square root of its second-order moment goes to zero at least as fast as $1/N^\alpha$, as N tends to infinity.

First, we establish the asymptotic variance/covariance of $\hat{\phi}_p(\omega)$ in the case of *Gaussian complex/circular white noise*. The following result holds:

$$\boxed{\lim_{N\to\infty} E\left\{[\hat{\phi}_p(\omega_1) - \phi(\omega_1)][\hat{\phi}_p(\omega_2) - \phi(\omega_2)]\right\} = \begin{cases}\phi^2(\omega_1), & \omega_1 = \omega_2 \\ 0, & \omega_1 \neq \omega_2\end{cases}} \tag{2.4.21}$$

Note that, for white noise, $\phi(\omega) = \sigma^2$ (for all ω). Since $\lim_{N\to\infty} E\{\hat{\phi}_p(\omega)\} = \phi(\omega)$ (*cf.* the analysis in the previous subsection), in order to prove (2.4.21) it suffices to show that

$$\lim_{N\to\infty} E\left\{\hat{\phi}_p(\omega_1)\hat{\phi}_p(\omega_2)\right\} = \phi(\omega_1)\phi(\omega_2) + \phi^2(\omega_1)\delta_{\omega_1,\omega_2} \tag{2.4.22}$$

From (2.2.1), we obtain

$$E\left\{\hat{\phi}_p(\omega_1)\hat{\phi}_p(\omega_2)\right\} = \frac{1}{N^2}\sum_{t=1}^{N}\sum_{s=1}^{N}\sum_{p=1}^{N}\sum_{m=1}^{N} E\left\{e(t)e^*(s)e(p)e^*(m)\right\} \cdot e^{-i\omega_1(t-s)}e^{-i\omega_2(p-m)} \tag{2.4.23}$$

For general random processes, the evaluation of the expectation in (2.4.23) is relatively complicated. However, the following general result for Gaussian random variables can be used: If a, b, c, and d are jointly Gaussian (complex or real) random variables, then

$$\boxed{\begin{aligned} E\{abcd\} = E\{ab\}E\{cd\} + E\{ac\}E\{bd\} + E\{ad\}E\{bc\} \\ -2E\{a\}E\{b\}E\{c\}E\{d\} \end{aligned}} \tag{2.4.24}$$

For a proof of (2.4.24), see, for example, [Janssen and Stoica 1988] and references therein. Thus, if the white noise $e(t)$ is Gaussian as assumed, the fourth-order moment in (2.4.23) is found to be

$$\begin{aligned}E\left\{e(t)e^*(s)e(p)e^*(m)\right\} &= \left[E\left\{e(t)e^*(s)\right\}\right]\left[E\left\{e(p)e^*(m)\right\}\right]\\ &\quad + \left[E\left\{e(t)e(p)\right\}\right]\left[E\left\{e(s)e(m)\right\}\right]^*\\ &\quad + \left[E\left\{e(t)e^*(m)\right\}\right]\left[E\left\{e^*(s)e(p)\right\}\right]\\ &= \sigma^4(\delta_{t,s}\delta_{p,m} + \delta_{t,m}\delta_{s,p}) \qquad (2.4.25)\end{aligned}$$

Inserting (2.4.25) in (2.4.23) gives

$$\begin{aligned}E\left\{\hat{\phi}_p(\omega_1)\hat{\phi}_p(\omega_2)\right\} &= \sigma^4 + \frac{\sigma^4}{N^2}\sum_{t=1}^{N}\sum_{s=1}^{N} e^{-i(\omega_1-\omega_2)(t-s)}\\ &= \sigma^4 + \frac{\sigma^4}{N^2}\left|\sum_{t=1}^{N} e^{i(\omega_1-\omega_2)t}\right|^2\\ &= \sigma^4 + \frac{\sigma^4}{N^2}\left\{\frac{\sin[(\omega_1-\omega_2)N/2]}{\sin[(\omega_1-\omega_2)/2]}\right\}^2 \qquad (2.4.26)\end{aligned}$$

The limit of the second term in (2.4.26) is σ^4 when $\omega_1 = \omega_2$ and zero otherwise, and (2.4.22) follows at once.

Remark: Note that, in the previous case, it was indeed possible to derive the *finite-sample* variance of $\hat{\phi}_p(\omega)$. For colored noise, the preceding derivation becomes more difficult, and a different approach (presented shortly) is needed. See Exercise 2.13 for yet another approach that applies to general Gaussian signals. ■

Next, we consider the case of a much more general signal obtained by linear filtering of the Gaussian white-noise sequence $\{e(t)\}$ examined earlier. This signal is given by

$$y(t) = \sum_{k=1}^{\infty} h_k e(t-k) \qquad (2.4.27)$$

and its PSD is given by

$$\phi_y(\omega) = |H(\omega)|^2\phi_e(\omega) \qquad (2.4.28)$$

(*cf.* (1.4.9)). Here, $H(\omega) = \sum_{k=1}^{\infty} h_k e^{-i\omega k}$. The intermediate result that follows, concerned with signals of the foregoing type, appears to have an independent interest. (We will omit the index "p" of $\hat{\phi}_p(\omega)$ in order to simplify the notation.)

For $N \gg 1$,

$$\boxed{\hat{\phi}_y(\omega) = |H(\omega)|^2 \hat{\phi}_e(\omega) + \mathcal{O}(1/\sqrt{N})} \tag{2.4.29}$$

Hence, the periodograms approximately satisfy an equation of the form of (2.4.28) that is satisfied by the true PSDs.

In order to prove (2.4.29), first observe that

$$\begin{aligned}
\frac{1}{\sqrt{N}} \sum_{t=1}^{N} y(t) e^{-i\omega t} &= \frac{1}{\sqrt{N}} \sum_{t=1}^{N} \sum_{k=1}^{\infty} h_k e(t-k) e^{-i\omega(t-k)} e^{-i\omega k} \\
&= \frac{1}{\sqrt{N}} \sum_{k=1}^{\infty} h_k e^{-i\omega k} \sum_{p=1-k}^{N-k} e(p) e^{-i\omega p} \\
&= \frac{1}{\sqrt{N}} \sum_{k=1}^{\infty} h_k e^{-i\omega k} \\
&\quad \cdot \left[\sum_{p=1}^{N} e(p) e^{-i\omega p} + \sum_{p=1-k}^{0} e(p) e^{-i\omega p} - \sum_{p=N-k+1}^{N} e(p) e^{-i\omega p} \right] \\
&\triangleq H(\omega) \left[\frac{1}{\sqrt{N}} \sum_{p=1}^{N} e(p) e^{-i\omega p} \right] + \rho(\omega)
\end{aligned} \tag{2.4.30}$$

where

$$\begin{aligned}
\rho(\omega) &= \frac{1}{\sqrt{N}} \sum_{k=1}^{\infty} h_k e^{-i\omega k} \left[\sum_{p=1-k}^{0} e(p) e^{-i\omega p} - \sum_{p=N-k+1}^{N} e(p) e^{-i\omega p} \right] \\
&\triangleq \frac{1}{\sqrt{N}} \sum_{k=1}^{\infty} h_k e^{-i\omega k} \varepsilon_k(\omega)
\end{aligned} \tag{2.4.31}$$

Next, note that

$$\begin{aligned}
E\left\{\varepsilon_k(\omega)\right\} &= 0, \\
E\left\{\varepsilon_k(\omega)\varepsilon_j(\omega)\right\} &= 0 \text{ for all } k \text{ and } j, \text{ and} \\
E\left\{\varepsilon_k(\omega)\varepsilon_j^*(\omega)\right\} &= 2\sigma^2 \min(k, j)
\end{aligned}$$

which imply

$$E\{\rho(\omega)\} = 0, \qquad E\left\{\rho^2(\omega)\right\} = 0$$

and

$$
\begin{aligned}
E\left\{|\rho(\omega)|^2\right\} &= \frac{1}{N}\left|\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} h_k e^{-i\omega k} h_j^* e^{i\omega j} E\left\{\varepsilon_k(\omega)\varepsilon_j^*(\omega)\right\}\right| \\
&= \frac{2\sigma^2}{N}\left|\sum_{k=1}^{\infty} h_k e^{-i\omega k}\left\{\sum_{j=1}^{k} h_j^* e^{i\omega j} j + \sum_{j=k+1}^{\infty} h_j^* e^{i\omega j} k\right\}\right| \\
&\leq \frac{2\sigma^2}{N}\sum_{k=1}^{\infty}|h_k|\left\{\sum_{j=1}^{\infty}|h_j| j + \sum_{j=1}^{\infty}|h_j| k\right\} \\
&= \frac{4\sigma^2}{N}\left(\sum_{k=1}^{\infty}|h_k|\right)\left(\sum_{j=1}^{\infty}|h_j| j\right)
\end{aligned}
$$

If $\sum_{k=1}^{\infty} k|h_k|$ is finite (which is true, for example, if $\{h_k\}$ is exponentially stable; see [Söderström and Stoica 1989]), we have

$$
E\left\{|\rho(\omega)|^2\right\} \leq \frac{\text{constant}}{N} \tag{2.4.32}
$$

Now, from (2.4.30), we obtain

$$
\hat{\phi}_y(\omega) = |H(\omega)|^2\hat{\phi}_e(\omega) + \gamma(\omega) \tag{2.4.33}
$$

where

$$
\gamma(\omega) = H^*(\omega)E^*(\omega)\rho(\omega) + H(\omega)E(\omega)\rho^*(\omega) + \rho(\omega)\rho^*(\omega)
$$

and where

$$
E(\omega) = \frac{1}{\sqrt{N}}\sum_{t=1}^{N} e(t)e^{-i\omega t}
$$

Both $E(\omega)$ and $\rho(\omega)$ are linear combinations of Gaussian random variables, so they are also Gaussian distributed. This means that the fourth-order moment formula (2.4.24) can be used to obtain the second-order moment of $\gamma(\omega)$. By doing so, and also by using (2.4.32) and the fact that, for example, (see (2.4.32)),

$$
\begin{aligned}
\left|E\left\{\rho(\omega)E^*(\omega)\right\}\right| &\leq \left[E\left\{|\rho(\omega)|^2\right\}\right]^{1/2}\left[E\left\{|E(\omega)|^2\right\}\right]^{1/2} \\
&= \frac{\text{constant}}{\sqrt{N}}\cdot\left[E\left\{|\hat{\phi}_e(\omega)|^2\right\}\right]^{1/2} = \frac{\text{constant}}{\sqrt{N}}
\end{aligned}
$$

we can verify that $\gamma(\omega) = \mathcal{O}(1/\sqrt{N})$, and hence the proof of (2.4.29) is concluded.

The main result of this section is derived by combining (2.4.21) and (2.4.29):

The asymptotic variance/covariance result (2.4.21) is also valid for a general linear signal as defined in (2.4.27). (2.4.34)

Remark: In the introduction to Chapter 1, we mentioned that the analysis of a complex-valued signal is not always more general than the analysis of the corresponding real-valued signal; we supported this claim by the example of a complex sine wave. Here, we have another instance where the claim is valid. Much as in the complex sinusoidal signal case, the complex (or circular) white noise does not specialize, in a direct manner, to real-valued white noise. Indeed, if we would let $e(t)$ in (2.4.19) be real valued, then the two equations in (2.4.19) would conflict with each other (for $t = s$). The *real-valued white noise random process* is a stationary signal that satisfies

$$\boxed{E\left\{e(t)e(s)\right\} = \sigma^2\delta_{t,s}} \tag{2.4.35}$$

If we try to carry out the proof of (2.4.21) under (2.4.35), then we find that the proof has to be modified. This was expected: both $\phi(\omega)$ and $\hat{\phi}_p(\omega)$ are even functions in the real-valued case; hence, (2.4.21) should be modified to include the case of both $\omega_1 = \omega_2$ and $\omega_1 = -\omega_2$. ■

It follows from (2.4.34) that, for a fairly general class of signals, the periodogram values are asymptotically (for $N \gg 1$) uncorrelated random variables whose means and standard deviations are both equal to the corresponding true PSD values. Hence, the periodogram is an *inconsistent spectral estimator*, which continues to fluctuate around the true PSD, with a nonzero variance, even if the length of the processed sample increases without bound. Furthermore, the fact that the periodogram values $\hat{\phi}_p(\omega)$ are uncorrelated (for large N values) makes the periodogram exhibit an *erratic behavior* (similar to that of a white-noise realization). These facts constitute the main limitations of the periodogram approach to PSD estimation. In the next sections, we present several modified periodogram-based methods that attempt to cure the aforementioned difficulties of the basic periodogram approach. As we shall see, the "improved methods" decrease the variance of the estimated spectrum at the expense of increasing its bias (and, hence, decreasing the average resolution).

2.5 THE BLACKMAN–TUKEY METHOD

In this section, we develop the Blackman–Tukey method [Blackman and Tukey 1959] and compare it to the periodogram. In later sections, we consider several other refined periodogram-based methods that, like the Blackman–Tukey (BT) method, seek to reduce the statistical variability of the estimated spectrum; we will compare these methods with one another and with the Blackman–Tukey method.

2.5.1 The Blackman–Tukey Spectral Estimate

As we have seen, the main problem with the periodogram is the high statistical variability of this spectral estimator, even for very large sample lengths. The poor statistical quality of the periodogram PSD estimator has been intuitively explained as arising from both the poor accuracy of $\hat{r}(k)$ in $\hat{\phi}_c(\omega)$ for extreme lags ($|k| \simeq N$) and the large number of (even if small) covariance estimation errors that are cumulatively summed up in $\hat{\phi}_c(\omega)$. Both these effects may be reduced by truncating the sum in the definition formula of $\hat{\phi}_c(\omega)$, (2.2.2). Following this idea leads to the Blackman–Tukey estimator, which is given by

$$\boxed{\hat{\phi}_{BT}(\omega) = \sum_{k=-(M-1)}^{M-1} w(k)\hat{r}(k)e^{-i\omega k}} \tag{2.5.1}$$

where $\{w(k)\}$ is an even function (i.e., $w(-k) = w(k)$), $w(0) = 1$, $w(k) = 0$ for $|k| \geq M$, and $w(k)$ decays smoothly to zero with k, and where $M < N$. Since $w(k)$ in (2.5.1) weights the lags of the sample covariance sequence, it is called a *lag window*.

If $w(k)$ in (2.5.1) is selected as the rectangular lag window, (i.e., $w(k) = 1$), then we simply obtain a truncated version of $\hat{\phi}_c(\omega)$. However, we may choose $w(k)$ in many other ways, and this flexibility may be employed to improve the accuracy of the Blackman–Tukey spectral estimator or to emphasize some of its characteristics that are of particular interest in a given application. In the next subsections, we address the principal issues that concern the problem of window selection. However, before doing so, we rewrite (2.5.1) in an alternative form that will be used in several places in the discussion that follows.

Let $W(\omega)$ denote the DTFT of $w(k)$,

$$W(\omega) = \sum_{k=-\infty}^{\infty} w(k)e^{-i\omega k} = \sum_{k=-(M-1)}^{M-1} w(k)e^{-i\omega k} \tag{2.5.2}$$

Making use of the DTFT property that led to (2.4.8), we can then write

$$\begin{aligned}
\hat{\phi}_{BT}(\omega) &= \sum_{k=-\infty}^{\infty} w(k)\hat{r}(k)e^{-i\omega k} \\
&= \text{DTFT of the product of the sequences} \\
&\quad \{\ldots, 0, 0, w(-(M-1)), \ldots, w(M-1), 0, 0, \ldots\} \text{ and} \\
&\quad \{\ldots, 0, 0, \hat{r}(-(N-1)), \ldots, \hat{r}(N-1), 0, 0, \ldots\} \\
&= \{\text{DTFT}(\hat{r}(k))\} * \{\text{DTFT}(w(k))\}
\end{aligned}$$

As DTFT$\{\ldots, 0, 0, \hat{r}(-(N-1)), \ldots, \hat{r}(N-1), 0, 0, \ldots\} = \hat{\phi}_p(\omega)$, we obtain

$$\boxed{\hat{\phi}_{BT}(\omega) = \hat{\phi}_p(\omega) * W(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{\phi}_p(\psi)W(\omega - \psi)d\psi} \tag{2.5.3}$$

This equation is analogous to (2.4.8) and can be interpreted in the same way. Since, for most windows in common use, $W(\omega)$ has a dominant, relatively narrow peak at $\omega = 0$, it follows from (2.5.3) that

The Blackman–Tukey spectral estimator (2.5.1) corresponds to a "locally" weighted average of the periodogram. (2.5.4)

Since the function $W(\omega)$ in (2.5.3) acts as a window (or weighting) in the frequency domain, it is sometimes called a *spectral window*. As we shall see, several refined periodogram-based spectral estimators discussed in what follows can be given an interpretation similar to that afforded by (2.5.3).

The form (2.5.3) in which the Blackman–Tukey spectral estimator has been cast is quite appealing from an intuitive standpoint. The main problem with the periodogram lies in its large variations about the true PSD. The weighted average in (2.5.3), in the neighborhood of the current frequency point ω, should smooth the periodogram and hence eliminate its large fluctuations.

On the other hand, this smoothing by the spectral window $W(\omega)$ will also have the undesirable effect of reducing the resolution. We may expect that the smaller the M, the larger the reduction in variance and the lower the resolution. These qualitative arguments may be made exact by a statistical analysis of $\hat{\phi}_{BT}(\omega)$, similar to that in the previous section. In fact, it is clear from (2.5.3) that the mean and variance of $\hat{\phi}_{BT}(\omega)$ can be derived from those of $\hat{\phi}_p(\omega)$. Roughly speaking, the results that can be established by the analysis of $\hat{\phi}_{BT}(\omega)$, based on (2.5.3), show that the resolution of this spectral estimator is on the order of $1/M$, whereas its variance is on the order of M/N. The compromise between resolution and variance, which should be considered when choosing the window's length, is clearly seen from the preceding considerations. We will look at the resolution-variance tradeoff in more detail in what follows. The next discussion addresses some of the main issues that concern window design.

2.5.2 Nonnegativeness of the Blackman–Tukey Spectral Estimate

Since $\phi(\omega) \geq 0$, it is natural to also require that $\hat{\phi}_{BT}(\omega) \geq 0$. The lag window can be selected to achieve this desirable property of the estimated spectrum. The following result holds true:

If the lag window $\{w(k)\}$ is positive semidefinite (i.e., $W(\omega) \geq 0$), then the windowed covariance sequence $\{w(k)\hat{r}(k)\}$ (with $\hat{r}(k)$ given by (2.2.4)) is positive semidefinite, too; this result implies that $\hat{\phi}_{BT}(\omega) \geq 0$ for all ω. (2.5.5)

In order to prove the previous result, first note that $\hat{\phi}_{BT}(\omega) \geq 0$ if and only if the sequence $\{\ldots, 0, 0, w(-(M-1))\hat{r}(-(M-1)), \ldots, w(M-1)\hat{r}(M-1), 0, 0, \ldots\}$ is positive semidefinite or, equivalently, the following Toeplitz matrix is positive semidefinite for all dimensions:

$$\begin{bmatrix} w(0)\hat{r}(0) & \ldots & w(M-1)\hat{r}(M-1) & & 0 \\ \vdots & & & \ddots & \\ w(-M+1)\hat{r}(-M+1) & & \ddots & & w(M-1)\hat{r}(M-1) \\ & \ddots & & & \vdots \\ 0 & & w(-M+1)\hat{r}(-M+1) & \ldots & w(0)\hat{r}(0) \end{bmatrix} =$$

$$\begin{bmatrix} w(0) & \ldots & w(M-1) & & 0 \\ \vdots & & & \ddots & \\ w(-M+1) & & \ddots & & w(M-1) \\ & \ddots & & & \vdots \\ 0 & & w(-M+1) & \ldots & w(0) \end{bmatrix} \odot \begin{bmatrix} \hat{r}(0) & \ldots & \hat{r}(M-1) & & 0 \\ \vdots & & & \ddots & \\ \hat{r}(-M+1) & & \ddots & & \hat{r}(M-1) \\ & \ddots & & & \vdots \\ 0 & & \hat{r}(-M+1) & \ldots & \hat{r}(0) \end{bmatrix}$$

The symbol $\odot$ denotes the Hadamard matrix product (i.e., element-wise multiplication). By a result in matrix theory, the Hadamard product of two positive semidefinite matrices is also a positive semidefinite matrix. (See Result R19 in Appendix A.) Thus, the proof of (2.5.5) is concluded.

Another, perhaps simpler, proof of (2.5.5) makes use of (2.5.3) in the following way: Since the sequence $\{w(k)\}$ is real and symmetric about the point $k = 0$, its DTFT $W(\omega)$ is an even, real-valued function. Furthermore, if $\{w(k)\}$ is a positive semidefinite sequence, then $W(\omega) \geq 0$ for all ω values. (See Exercise 1.8.) By (2.5.3), $W(\omega) \geq 0$ immediately implies $\hat{\phi}_{BT}(\omega) \geq 0$, as $\hat{\phi}_p(\omega) \geq 0$ by definition.

On one hand, it should be noted that some lag windows, such as the rectangular window, do not satisfy the assumption made in (2.5.5); hence, their use could lead to estimated spectra that take negative values. The Bartlett window, on the other hand, is positive semidefinite (as can be seen from (2.4.15)).

2.6 WINDOW DESIGN CONSIDERATIONS

The properties of the Blackman–Tukey estimator (and of other refined periodogram methods discussed in the next section) are related directly to the choice of the lag window. In this section, we discuss several relevant properties of windows that are useful in selecting or designing a window to use in a refined spectral estimation procedure.

2.6.1 Time-Bandwidth Product and Resolution-Variance Tradeoffs in Window Design

Most windows are such that they take only nonnegative values in both time and frequency domains (or, if they also take negative values, these are much smaller than the positive values of the window). In addition, they peak at the origin in both domains. For this type of window, it is possible to define an *equivalent time width*, N_e, and an *equivalent bandwidth*, β_e, as follows:

$$N_e = \frac{\sum_{k=-(M-1)}^{M-1} w(k)}{w(0)} \tag{2.6.1}$$

and

$$\beta_e = \frac{\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)d\omega}{W(0)} \tag{2.6.2}$$

From the definitions of direct and inverse DTFTs, we obtain

$$W(0) = \sum_{k=-\infty}^{\infty} w(k) = \sum_{k=-(M-1)}^{M-1} w(k) \tag{2.6.3}$$

and

$$w(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)d\omega \tag{2.6.4}$$

Using (2.6.3) and (2.6.4) in (2.6.1) and (2.6.2) gives the following result:

> The (equivalent) time-bandwidth product equals unity:
> $$N_e\beta_e = 1 \tag{2.6.5}$$

As already indicated, the preceding result applies to window-like signals. Some extended results of the time-bandwidth product type, which apply to more general classes of signals, are presented in Complement 2.8.5.

It is clearly seen from (2.6.5) that a window cannot be both time limited and bandlimited. The more slowly the window decays to zero in one domain, the more concentrated it is in the other domain. The simple result (2.6.5) has several other interesting consequences, as explained next.

The equivalent temporal extent (or aperture), N_e, of $w(k)$ is essentially determined by the window's length. For example, for a rectangular window, we have $N_e \simeq 2M$, whereas for a triangular window, $N_e \simeq M$. This observation, together with (2.6.5), implies that the equivalent bandwidth β_e is basically determined by the window's length. More precisely, $\beta_e = \mathcal{O}(1/M)$. This fact lends support to a claim made previously that, for a window that concentrates most of its energy in its main lobe, the width of that lobe should be on the order of $1/M$. Since the main lobe's width sets a limit on the spectral resolution achievable (as explained in Section 2.4), the preceding observation shows that the spectral resolution limit of a windowed method should be on the order of $1/M$. On the other hand, as explained in the previous section, the statistical variance of such a method is essentially proportional to M/N. Hence, we reach the following conclusion:

> The choice of window's length should be based on a tradeoff between spectral resolution and statistical variance. (2.6.6)

As a rule of thumb, we should choose $M \leq N/10$ in order to reduce the standard deviation of the estimated spectrum by at least a factor of three as compared with the periodogram.

Once M is determined, we cannot decrease simultaneously the energy in the main lobe (to reduce smearing) and the energy in the sidelobes (to reduce leakage). This follows, for example, from (2.6.4), which shows that the area of $W(\omega)$ is fixed once $w(0)$ is fixed (such as $w(0) = 1$). In other words, if we want to decrease the main lobe's width then we should accept an increase in the sidelobe energy and vice versa. In summary,

> The selection of window's shape should be based on a tradeoff between smearing and leakage effects. (2.6.7)

The tradeoff just mentioned is usually dictated by the specific application at hand. A number of windows have been developed to address this tradeoff. In some sense, each of these windows can be seen as a design at a specific point in the resolution–leakage tradeoff curve. We consider several such windows in the next subsection.

2.6.2 Some Common Lag Windows

In this section, we list some of the most common lag windows and outline their relevant properties. Our purpose is not to provide a detailed derivation or an exhaustive listing of such windows, but rather to provide a quick reference of common windows. More detailed information on these and other windows can be found in [Harris 1978; Kay 1988; Marple 1987; Oppenheim and Schafer 1989; Priestley 1981; Porat 1997], where many of the closed-form windows have been compiled. Table 2.1 lists some common windows along with some useful properties.

In addition to the fixed-window designs in Table 2.1, there are windows that contain a design parameter that may be varied to trade between resolution and sidelobe leakage. Two such common designs are the Chebyshev window and the Kaiser window. The Chebyshev window has the property that the peak level of the sidelobe "ripples" is constant. Thus, unlike in most other windows, the sidelobe level does not decrease as ω increases. The Kaiser window is defined by

$$w(k) = \frac{I_0\left(\gamma\sqrt{1 - [k/(M-1)]^2}\right)}{I_0(\gamma)}, \quad -(M-1) \leq k \leq M-1 \tag{2.6.8}$$

where $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. The parameter γ trades the main lobe width for the sidelobe leakage level; $\gamma = 0$ corresponds to a rectangular window, and $\gamma > 0$ results in lower sidelobe leakage at the expense of a broader main lobe. The approximate value of γ needed to achieve a peak sidelobe level of B dB below the peak value is

$$\gamma \simeq \begin{cases} 0, & B < 21 \\ 0.584(B-21)^{0.4} + 0.0789(B-21), & 21 \leq B \leq 50 \\ 0.11(B-8.7), & B > 50 \end{cases}$$

Table 2.1 Some Common Windows and their Properties
These windows satisfy $w(k) \equiv 0$ for $|k| \geq M$, and $w(k) = w(-k)$; the defining equations are valid for $0 \leq k \leq (M - 1)$.

Window Name	Defining Equation	Approx. Main Lobe Width (radians)	Sidelobe Level (dB)
Rectangular	$w(k) = 1$	$2\pi/M$	-13
Bartlett	$w(k) = \frac{M-k}{M}$	$4\pi/M$	-25
Hanning	$w(k) = 0.5 + 0.5\cos\left(\frac{\pi k}{M}\right)$	$4\pi/M$	-31
Hamming	$w(k) = 0.54 + 0.46\cos\left(\frac{\pi k}{M-1}\right)$	$4\pi/M$	-41
Blackman	$w(k) = 0.42 + 0.5\cos\left(\frac{\pi k}{M-1}\right)$ $+ 0.08\cos\left(\frac{\pi k}{M-1}\right)$	$6\pi/M$	-57

The Kaiser window is an approximation of the optimal window described in the next subsection. It is often chosen over the fixed-window designs because it has a lower sidelobe level when γ is selected to give the same main lobe width as the corresponding fixed window (or narrower main lobe width for a given sidelobe level). The optimal window of the next subsection improves on the Kaiser design slightly.

Figure 2.3 shows plots of several windows with $M = 26$. The Kaiser window is shown for $\gamma = 1$ and $\gamma = 4$, and the Chebyshev window is designed to have a -40 dB sidelobe level. Figure 2.4 shows the corresponding normalized window transfer functions $W(\omega)$. Note the constant sidelobe ripple level of the Chebyshev design.

We remark that, except for the Bartlett window, none of the windows we have introduced (including the Chebyshev and Kaiser windows) has nonnegative Fourier transform. On the other hand, it is straightforward to produce such a nonnegative definite window by convolving the window with itself. Recall that the Bartlett window is the convolution of a rectangular window with itself. We will make use of the convolution of windows with themselves in the next two subsections, both for window design and for relating temporal windows to covariance lag windows.

2.6.3 Window Design Example

Assume a situation where the observed signal consists of a weak desired signal and a strong interference and where both the desired signal and the interference are narrowband signals that are well separated in frequency. However, there is no *a priori* quantitative information available on the frequency separation between the desired signal and the interference. It is required to design a lag window for use in a Blackman–Tukey spectral estimation method, with the purpose of detecting and locating in frequency the useful signal.

The main problem in the application just outlined lies in the fact that the (strong) interference might completely mask the (weak) desired signal, through leakage. In order to get rid of this

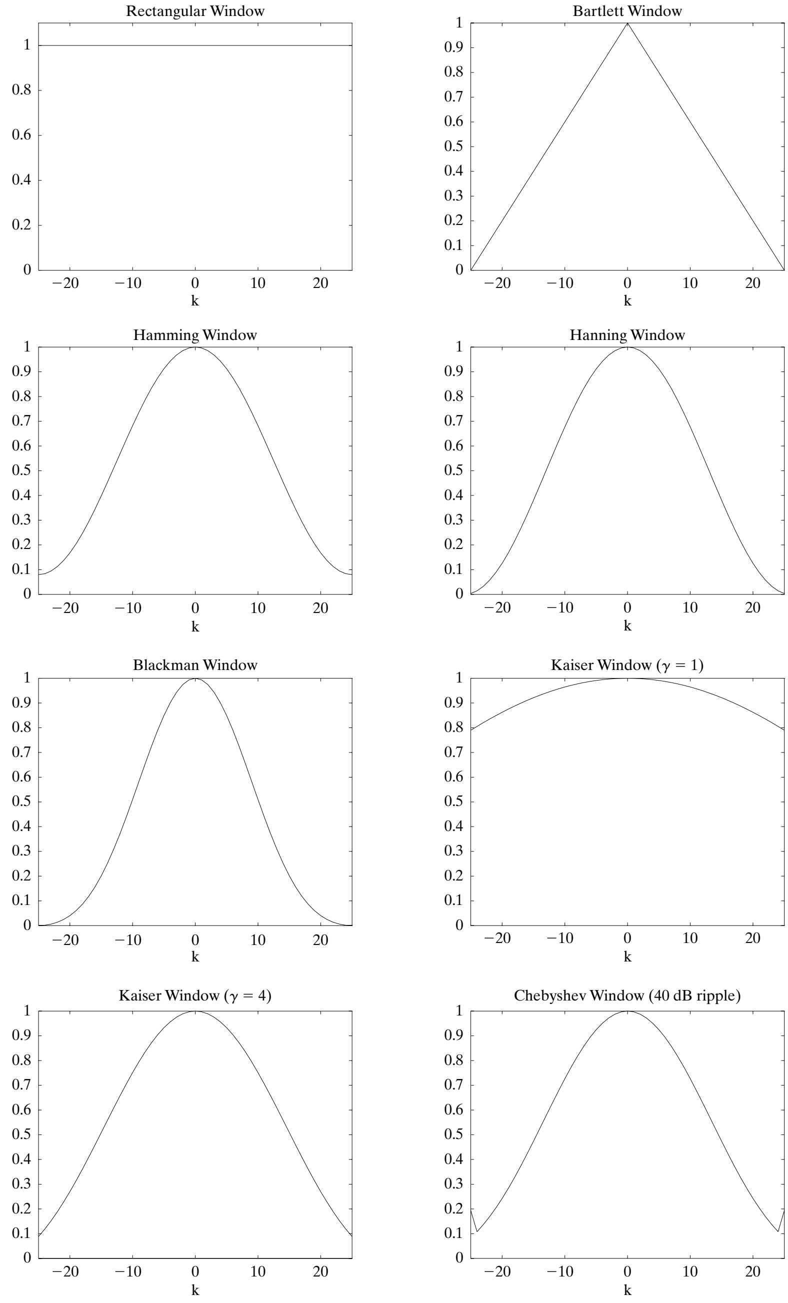
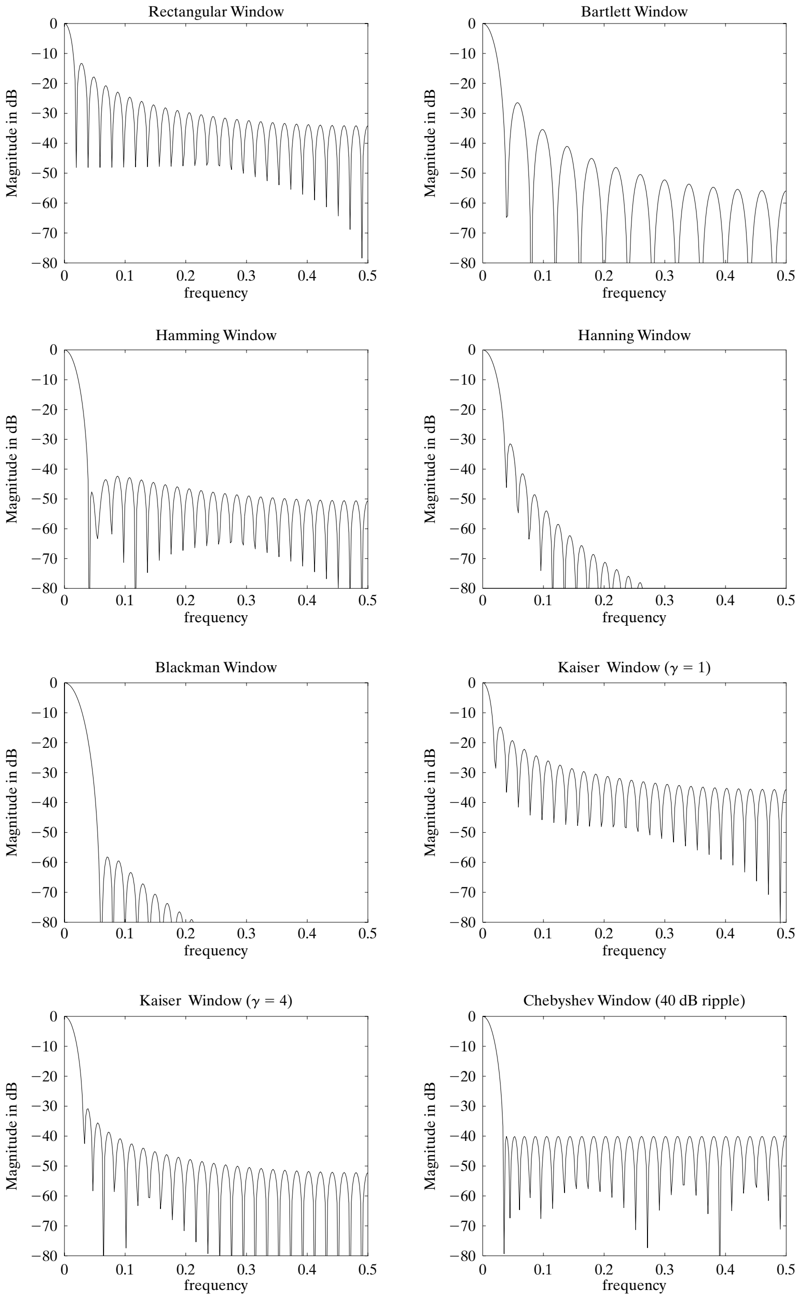


Figure 2.3 Some common window functions (shown for $M = 26$). The Kaiser window uses $\gamma = 1$ and $\gamma = 4$, and the Chebyshev window is designed for a -40 dB sidelobe level.



Rectangular Window

Bartlett Window

Hamming Window

Hanning Window

Blackman Window

Kaiser Window ($\gamma = 1$)

Kaiser Window ($\gamma = 4$)

Chebyshev Window (40 dB ripple)

Figure 2.4 The DTFTs of the window functions in Figure 2.3.

problem, the window design should compromise smearing for leakage. Note that the smearing effect is not of main concern in this application, because the useful signal and the interference are well separated in frequency. Hence, smearing will not affect our ability to *detect* the desired signal although it will limit, to some degree, our ability to accurately *locate in frequency* the signal in question.

We consider a window sequence whose DTFT $W(\omega)$ is constructed as the squared magnitude of the DTFT of another sequence $\{v(k)\}$; in this way, we guarantee that the constructed window is positive semidefinite. Mathematically, the previous design problem can be formulated as follows: Consider a sequence $\{v(0), \ldots, v(M-1)\}$, and let

$$V(\omega) = \sum_{k=0}^{M-1} v(k) e^{-i\omega k} \tag{2.6.9}$$

The DTFT $V(\omega)$ can be rewritten in the more compact form

$$V(\omega) = v^* a(\omega) \tag{2.6.10}$$

where

$$v = [v(0) \ \ldots \ v(M-1)]^* \tag{2.6.11}$$

and

$$a(\omega) = [1 \ e^{-i\omega} \ldots \ e^{-i(M-1)\omega}]^T \tag{2.6.12}$$

Define the spectral window as

$$W(\omega) = |V(\omega)|^2 \tag{2.6.13}$$

The corresponding lag window can be obtained from (2.6.13) as follows:

$$\begin{aligned}\sum_{k=-(M-1)}^{M-1} w(k) e^{-i\omega k} &= \sum_{n=0}^{M-1} \sum_{p=0}^{M-1} v(n) v^*(p) e^{-i\omega(n-p)} \\ &= \sum_{n=0}^{M-1} \sum_{k=n}^{n-(M-1)} v(n) v^*(n-k) e^{-i\omega k} \\ &= \sum_{k=-(M-1)}^{M-1} \left[\sum_{n=0}^{M-1} v(n) v^*(n-k) \right] e^{-i\omega k}\end{aligned} \tag{2.6.14}$$

This gives

$$w(k) = \sum_{n=0}^{M-1} v(n) v^*(n-k) \tag{2.6.15}$$

The last equality in (2.6.14), and hence the equality (2.6.15), are valid under the convention that $v(k) = 0$ for $k < 0$ and $k \geq M$.

As already mentioned, this method of constructing $\{w(k)\}$ from the convolution of the sequence $\{v(k)\}$ with itself has the advantage that the lag window so obtained is always positive semidefinite or, equivalently, the corresponding spectral window satisfies $W(\omega) \geq 0$ (as is easily seen from (2.6.13)). Besides this, the design of $\{w(k)\}$ can be reduced to the selection of $\{v(k)\}$, which may be more conveniently done, as explained next.

In the present application, the design objective is to reduce the leakage incurred by $\{w(k)\}$ as much as possible. This objective can be formulated as the problem of minimizing the relative energy in the sidelobes of $W(\omega)$ or, equivalently, as the problem of maximizing the relative energy in the main lobe of $W(\omega)$:

$$\max_{v} \left\{ \frac{\int_{-\beta\pi}^{\beta\pi} W(\omega) d\omega}{\int_{-\pi}^{\pi} W(\omega) d\omega} \right\} \tag{2.6.16}$$

Here, β is a design parameter that quantifies how much smearing (or resolution) we can trade off for leakage reduction. The larger the β is, the more leakage free is the optimal window derived from (2.6.16), but also the more diminished is the spectral resolution associated with that window.

By writing the criterion in (2.6.16) in the form

$$\frac{\frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} |V(\omega)|^2 d\omega}{\frac{1}{2\pi}\int_{-\pi}^{\pi} |V(\omega)|^2 d\omega} = \frac{v^* \left[\frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} a(\omega)a^*(\omega) d\omega\right] v}{v^* v} \tag{2.6.17}$$

(*cf.* (2.6.10) and Parseval's theorem, (1.2.6)), the optimization problem (2.6.16) becomes

$$\max_{v} \frac{v^* \Gamma v}{v^* v} \tag{2.6.18}$$

where

$$\Gamma = \frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} a(\omega)a^*(\omega) d\omega \triangleq [\gamma_{m-n}] \tag{2.6.19}$$

and where

$$\gamma_{m-n} = \frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} e^{-i(m-n)\omega} d\omega = \frac{\sin[(m-n)\beta\pi]}{(m-n)\pi} \tag{2.6.20}$$

(note that $\gamma_0 = \beta$). By using the function

$$\operatorname{sinc}(x) \triangleq \frac{\sin x}{x}, \qquad (\operatorname{sinc}(0) = 1) \tag{2.6.21}$$

we can write (2.6.20) as

$$\boxed{\gamma_{m-n} = \beta \operatorname{sinc}[(m-n)\beta\pi]} \tag{2.6.22}$$

The solution to the problem (2.6.18) is well known: the maximizing v is given by the dominant eigenvector of Γ, associated with the maximum eigenvalue of this matrix. (See Result R13 in Appendix A.) To summarize,

> The optimal lag window that minimizes the relative energy in the sidelobe interval $[-\pi, -\beta\pi] \cup [\beta\pi, \pi]$ is given by (2.6.15), where v is the dominant eigenvector of the matrix Γ defined in (2.6.19) and (2.6.22). (2.6.23)

Regarding the choice of the design parameter β, it is clear that β should be larger than $1/M$ in order to allow for a significant reduction of leakage. Otherwise, by selecting, for example, $\beta \simeq 1/M$, we weight the resolution issue too much in the design problem, with unfavorable consequences for leakage reduction.

Finally, we remark that a problem quite similar to the previous problem, although derived from different considerations, will be encountered in Chapter 5. (See also [MULLIS AND SCHARF 1991].)

2.6.4 Temporal Windows and Lag Windows

As we have previously seen, the unwindowed periodogram coincides with the unwindowed correlogram. The Blackman–Tukey estimator is a windowed correlogram obtained by using a lag window. Similarly, we can define a windowed periodogram

$$\hat{\phi}_W(\omega) = \frac{1}{N}\left|\sum_{t=1}^{N} v(t)y(t)e^{-i\omega t}\right|^2 \tag{2.6.24}$$

where the weighting sequence $\{v(t)\}$ may be called a *temporal window*. A temporal window is sometimes called a *taper*. Welch [WELCH 1967] was one of the first researchers who considered windowed periodogram spectral estimators (see Section 2.7.2 for a description of Welch's method); hence, the subscript "W" is attached to $\hat{\phi}(\omega)$ in (2.6.24). However, while the reason for windowing the correlogram is clearly motivated, the reason for windowing the periodogram is less obvious. In order to motivate (2.6.24), at least partially, write it as

$$\hat{\phi}_W(\omega) = \frac{1}{N}\sum_{t=1}^{N}\sum_{s=1}^{N} v(t)v^*(s)y(t)y^*(s)e^{-i\omega(t-s)} \tag{2.6.25}$$

Next, take expectation of both sides of (2.6.25), to obtain

$$E\left\{\hat{\phi}_W(\omega)\right\} = \frac{1}{N}\sum_{t=1}^{N}\sum_{s=1}^{N} v(t)v^*(s)r(t-s)e^{-i\omega(t-s)} \tag{2.6.26}$$

Inserting

$$r(t-s) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi(\omega) e^{i\omega(t-s)} d\omega \tag{2.6.27}$$

in (2.6.26) gives

$$\begin{aligned} E\left\{\hat{\phi}_W(\omega)\right\} &= \frac{1}{N2\pi}\int_{-\pi}^{\pi} \phi(\psi)\left[\sum_{t=1}^{N}\sum_{s=1}^{N} v(t)v^*(s)e^{-i(\omega-\psi)(t-s)}\right] d\psi \\ &= \frac{1}{N2\pi}\int_{-\pi}^{\pi} \phi(\psi)\left|\sum_{t=1}^{N} v(t)e^{-i(\omega-\psi)t}\right|^2 d\psi \end{aligned} \tag{2.6.28}$$

Define

$$W(\omega) = \frac{1}{N}\left|\sum_{t=1}^{N} v(t)e^{-i\omega t}\right|^2 \tag{2.6.29}$$

By using this notation, we can write (2.6.28) as

$$E\left\{\hat{\phi}_W(\omega)\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \phi(\psi)W(\omega-\psi)d\psi \tag{2.6.30}$$

As the equation (2.6.29) is similar to (2.6.13), the sequence whose DTFT is equal to $W(\omega)$ immediately follows from (2.6.15):

$$\boxed{w(k) = \frac{1}{N}\sum_{n=1}^{N} v(n)v^*(n-k)} \tag{2.6.31}$$

Next, by comparing (2.6.30) and (2.5.3), we get the following result:

The windowed periodogram and the windowed correlogram have the same *average* behavior, provided that the temporal and lag windows are related as in (2.6.31). (2.6.32)

Hence, $E\{\hat{\phi}_W(\omega)\} = E\{\hat{\phi}_{BT}(\omega)\}$, provided that the temporal and lag windows are matched to one another. However, a similarly simple relationship between $\hat{\phi}_W(\omega)$ and $\hat{\phi}_{BT}(\omega)$ does not seem to exist. This makes it somewhat difficult to motivate the windowed periodogram as defined in (2.6.24). The Welch periodogram, though, does not weight all data samples as in (2.6.24) and is a useful spectral estimator (as is shown in the next section).

2.7 OTHER REFINED PERIODOGRAM METHODS

In Section 2.5, we introduced the Blackman–Tukey estimator as an alternative to the periodogram. In this section, we present three other modified periodograms: the Bartlett, Welch, and Daniell methods. Like the Blackman–Tukey method, they seek to reduce the variance of the periodogram by smoothing or averaging the periodogram estimates in some way. We will relate these methods to one another and to the Blackman–Tukey method.

2.7.1 Bartlett Method

The basic idea of the Bartlett method [Bartlett 1948; Bartlett 1950] is simple: to reduce the large fluctuations of the periodogram, split up the available sample of N observations into $L = N/M$ subsamples of M observations each, and then average the periodograms obtained from the subsamples for each value of ω. Mathematically, the Bartlett method can be described as follows: Let

$$y_j(t) = y((j-1)M + t), \qquad \begin{array}{rcl} t & = & 1, \ldots, M \\ j & = & 1, \ldots, L \end{array} \tag{2.7.1}$$

denote the observations of the jth subsample, and let

$$\hat{\phi}_j(\omega) = \frac{1}{M} \left| \sum_{t=1}^{M} y_j(t) e^{-i\omega t} \right|^2 \tag{2.7.2}$$

denote the corresponding periodogram. The Bartlett spectral estimate is then given by

$$\hat{\phi}_B(\omega) = \frac{1}{L} \sum_{j=1}^{L} \hat{\phi}_j(\omega) \tag{2.7.3}$$

Since the Bartlett method operates on data segments of length M, the resolution afforded should be on the order of $1/M$. Hence, the spectral resolution of the Bartlett method is reduced by a factor L, compared with the resolution of the original periodogram method. In return for this reduction in resolution, we can expect the Bartlett method to have a reduced variance. We show below that the Bartlett method reduces the variance of the periodogram by the same factor L. The compromise between resolution and variance when selecting M (or L) is thus evident.

An interesting way to look at the Bartlett method and its properties is by relating it to the Blackman–Tukey method. As we know, $\hat{\phi}_j(\omega)$ of (2.7.2) can be rewritten as

$$\hat{\phi}_j(\omega) = \sum_{k=-(M-1)}^{M-1} \hat{r}_j(k) e^{-i\omega k} \tag{2.7.4}$$

where $\{\hat{r}_j(k)\}$ is the sample covariance sequence corresponding to the jth subsample. Inserting (2.7.4) in (2.7.3) gives

$$\hat{\phi}_B(\omega) = \sum_{k=-(M-1)}^{M-1} \left[\frac{1}{L} \sum_{j=1}^{L} \hat{r}_j(k) \right] e^{-i\omega k} = \sum_{k=-(M-1)}^{M-1} \hat{r}_B(k) e^{-i\omega k} \tag{2.7.5}$$

We see that $\hat{\phi}_B(\omega)$ is similar in form to the Blackman–Tukey estimator that uses a rectangular window. Here, $\hat{r}_B(k)$ is an estimate of the ACS $r(k)$. However, $\hat{r}_B(k)$ in (2.7.5) does not make efficient use of the available lag products $y(t)y^*(t-k)$, especially for $|k|$ near $M-1$. (See Exercise 2.14.) In fact, for $k = M-1$, only about $1/M$th of the available lag products are used to form the ACS estimate in (2.7.5). We expect that the variance of $\hat{r}_B(k)$ will be higher than that of the corresponding $\hat{r}(k)$ lags used in the Blackman–Tukey estimate and, similarly, that the variance of $\hat{\phi}_B(\omega)$ will be higher than that of $\hat{\phi}_{BT}(\omega)$. In addition, the Bartlett method uses a fixed rectangular lag window and thus provides less flexibility in resolution–leakage tradeoff than does the Blackman–Tukey method. For these reasons, we conclude the following:

> The Bartlett estimate, as defined in (2.7.1)–(2.7.3), is similar in form to, but typically has a slightly higher variance than, the Blackman–Tukey estimate with a rectangular lag window of length M. (2.7.6)

The reduction in resolution and the decrease of variance (both by a factor $L = N/M$) for the Bartlett estimate, as compared to the basic periodogram method, follow from (2.7.6) and from the properties of the Blackman–Tukey spectral estimator given previously.

The main lobe of the rectangular window is narrower than that associated with most other lag windows—this follows from the observation that the rectangular window clearly has the largest equivalent time width and from the fact that the time-bandwidth product is constant; see (2.6.5). Thus, it follows from (2.7.6) that, in the class of Blackman–Tukey estimates, the Bartlett estimator can be expected to have the least smearing (and hence the best resolution), but the most significant leakage.

2.7.2 Welch Method

The Welch method [Welch 1967] is obtained by refining the Bartlett method in two respects. First, the data segments in the Welch method are allowed to overlap. Second, each data segment is windowed prior to computation of the periodogram. To describe the Welch method in a mathematical form, let

$$y_j(t) = y((j-1)K + t), \qquad \begin{array}{rcl} t &=& 1, \ldots, M \\ j &=& 1, \ldots, S \end{array} \tag{2.7.7}$$

denote the jth data segment. In (2.7.7), $(j-1)K$ is the starting point for the jth sequence of observations. If $K = M$, then the sequences do not overlap (but are contiguous), and we get the sample splitting used by the Bartlett method (which leads to $S = L = N/M$ data subsamples).

However, the value recommended for K in the Welch method is $K = M/2$, in which case $S \simeq 2M/N$ data segments (with 50% overlap between successive segments) are obtained.

The windowed periodogram corresponding to $y_j(t)$ is computed as

$$\hat{\phi}_j(\omega) = \frac{1}{MP} \left| \sum_{t=1}^{M} v(t) y_j(t) e^{-i\omega t} \right|^2 \tag{2.7.8}$$

where P denotes the "power" of the temporal window $\{v(t)\}$:

$$P = \frac{1}{M} \sum_{t=1}^{M} |v(t)|^2 \tag{2.7.9}$$

The Welch estimate is found by averaging the windowed periodograms in (2.7.8):

$$\hat{\phi}_W(\omega) = \frac{1}{S} \sum_{j=1}^{S} \hat{\phi}_j(\omega) \tag{2.7.10}$$

The reasons for these modifications to the Bartlett method, which led to the Welch method, are simple to explain. By allowing overlap between the data segments and hence getting more periodograms to be averaged in (2.7.10), we hope to decrease the variance of the estimated PSD. By introducing the window in the periodogram computation, we hope to get more control over the bias/resolution properties of the estimated PSD (see Section 2.6.4). Additionally, the temporal window may be used to give less weight to the data samples at the ends of each subsample, hence, making the consecutive subsample sequences less correlated to one another, even though they are overlapping. The principal effect of this "decorrelation" should be a more effective reduction of variance via the averaging in (2.7.10).

The analysis that led to the results (2.6.30)–(2.6.32) can be modified to show that the use of windowed periodograms in the Welch method, as contrasted to the unwindowed periodograms in the Bartlett method, indeed offers more flexibility in controlling the bias properties of the estimated spectrum. The variance of the Welch spectral estimator is more difficult to analyze (except in some special cases). However, there is empirical evidence that the Welch method can offer lower variance than the Bartlett method, but the difference in the variances corresponding to the two methods is not dramatic.

We can relate the Welch estimator to the Blackman–Tukey spectral estimator by a straightforward calculation, as we did for the Bartlett method. By inserting (2.7.8) in (2.7.10), we obtain

$$\begin{aligned} \hat{\phi}_W(\omega) &= \frac{1}{S} \sum_{j=1}^{S} \frac{1}{MP} \sum_{t=1}^{M} \sum_{k=1}^{M} v(t) v^*(k) y_j(t) y_j^*(k) e^{-i\omega(t-k)} \\ &= \frac{1}{MP} \sum_{t=1}^{M} \sum_{k=1}^{M} v(t) v^*(k) \underbrace{\left[\frac{1}{S} \sum_{j=1}^{S} y_j(t) y_j^*(k) \right]}_{\triangleq \tilde{r}(t,k)} e^{-i\omega(t-k)} \end{aligned} \tag{2.7.11}$$

For large values of N and for $K \le M/2$, S turns out to be sufficiently large for $\tilde{r}(t,k)$ to be a reasonable estimate of the covariance $r(t-k)$. We assume that $\tilde{r}(t,k)$ depends only on the difference $(t-k)$, at least approximately:

$$\tilde{r}(t,k) \simeq \tilde{r}(t-k) \tag{2.7.12}$$

Using (2.7.12) in (2.7.11) gives

$$\begin{aligned}\hat{\phi}_W(\omega) &\simeq \frac{1}{MP}\sum_{t=1}^{M}\sum_{k=1}^{M} v(t)v^*(k)\tilde{r}(t-k)e^{-i\omega(t-k)} \\ &= \frac{1}{MP}\sum_{t=1}^{M}\sum_{\tau=t-1}^{t-M} v(t)v^*(t-\tau)\tilde{r}(\tau)e^{-i\omega\tau} \\ &= \sum_{\tau=-(M-1)}^{M-1}\left[\frac{1}{MP}\sum_{t=1}^{M} v(t)v^*(t-\tau)\right]\tilde{r}(\tau)e^{-i\omega\tau} \end{aligned} \tag{2.7.13}$$

By introducing

$$w(\tau) = \frac{1}{MP}\sum_{t=1}^{M} v(t)v^*(t-\tau) \tag{2.7.14}$$

(under the convention that $v(k) = 0$ for $k < 1$ and $k > M$), we can write (2.7.13) as

$$\hat{\phi}_W(\omega) \simeq \sum_{\tau=-(M-1)}^{M-1} w(\tau)\tilde{r}(\tau)e^{-i\omega\tau} \tag{2.7.15}$$

which is to be compared with the form of the Blackman–Tukey estimator. To summarize, the Welch estimator has been shown to approximate a Blackman–Tukey-*type* estimator for the estimated covariance sequence (2.7.12) (which may be expected to have finite-sample properties different from those of $\hat{r}(k)$).

The Welch estimator can be efficiently computed via the FFT and is one of the most frequently used PSD estimation methods. Its previous interpretation is pleasing, even though approximate, because the Blackman–Tukey form of spectral estimator is theoretically the most favored one. This interpretation also shows that we may think of replacing the usual covariance estimates $\{\hat{r}(k)\}$ in the Blackman–Tukey estimator by other sample covariances, with the purpose of either reducing the computational burden or improving the statistical accuracy.

2.7.3 Daniell Method

As shown in (2.4.21), the periodogram values $\hat{\phi}(\omega_k)$ corresponding to different frequency values ω_k are (asymptotically) uncorrelated random variables. One may then think of reducing the large variance of the basic periodogram estimator by averaging the periodogram over small intervals centered on the current frequency ω. This is the idea behind the Daniell method [Daniell 1946]. The practical form of the Daniell estimate, which can be implemented by means of the FFT, is

$$\hat{\phi}_D(\omega_k) = \frac{1}{2J+1} \sum_{j=k-J}^{k+J} \hat{\phi}_p(\omega_j) \tag{2.7.16}$$

where

$$\omega_k = \frac{2\pi}{\tilde{N}} k, \qquad k = 0, \ldots, \tilde{N} - 1 \tag{2.7.17}$$

and where $\tilde{N}$ is (much) larger than N to ensure a fine sampling of $\hat{\phi}_p(\omega)$. The periodogram samples needed in (2.7.16) can be obtained, for example, by using a radix-2 FFT algorithm applied to the zero-padded data sequence, as described in Section 2.3. The parameter J in the Daniell method should be chosen sufficiently small to guarantee that $\phi(\omega)$ is nearly constant on the interval(s):

$$\left[\omega - \frac{2\pi}{\tilde{N}} J, \ \omega + \frac{2\pi}{\tilde{N}} J\right] \tag{2.7.18}$$

$\tilde{N}$ can, in principle, be chosen as large as we want, so we can choose J to be fairly large without violating the requirement that $\phi(\omega)$ be nearly constant over the interval in (2.7.18). For the sake of illustration, let us assume that we keep the ratio $J/\tilde{N}$ constant, but increase both J and $\tilde{N}$ significantly. As $J/\tilde{N}$ is constant, the resolution/bias properties of the Daniell estimator should be basically unaffected. On the other hand, the fact that the number of periodogram values averaged in (2.7.16) increases with increased J might suggest that the variance decreases. However, we know that this should not be possible, because the variance can be decreased only at the expense of increasing the bias (and vice versa). Indeed, in the case under discussion, the periodogram values averaged in (2.7.16) become more and more correlated as $\tilde{N}$ increases; hence, the variance of $\hat{\phi}_D(\omega)$ does not necessarily decrease with J if $\tilde{N}$ is larger than N. (See, for example, Exercise 2.13.) We will return to the bias and variance properties of the Daniell method a bit later.

By introducing $\beta = 2J/\tilde{N}$, one can write (2.7.18) in a form that is more convenient for the discussion that follows, namely

$$[\omega - \pi\beta, \omega + \pi\beta] \tag{2.7.19}$$

Equation (2.7.16) is a discrete approximation of the continuous version of the Daniell estimator, which is given by

$$\hat{\phi}_D(\omega) = \frac{1}{2\pi\beta} \int_{\omega-\beta\pi}^{\omega+\beta\pi} \hat{\phi}_p(\psi) d\psi \tag{2.7.20}$$

The larger the $\tilde{N}$, the smaller the difference between the approximation (2.7.16) and the continuous version (2.7.20) of the Daniell spectral estimator.

It is intuitively clear from (2.7.20) that, as β increases, the resolution of the Daniell estimator decreases (so the bias increases) and the variance decreases. In fact, if we introduce

$$M = 1/\beta \tag{2.7.21}$$

(in an approximate sense, as $1/\beta$ is not necessarily an integer) then we may expect that the resolution and the variance of the Daniell estimator are both decreased by a factor M, compared with the basic periodogram method. In order to support this claim, we relate the Daniell estimator to the Blackman–Tukey estimation technique. By comparing (2.7.20) and (2.5.3), we obtain the following result:

> The Daniell estimator is a particular case of the Blackman–Tukey class of spectral estimators, one corresponding to a rectangular *spectral* window:
>
> $$W(\omega) = \begin{cases} 1/\beta, & \omega \in [-\beta\pi, \beta\pi] \\ 0, & \text{otherwise} \end{cases}$$

(2.7.22)

This observation, along with the time-bandwidth product result and the properties of the Blackman–Tukey spectral estimator, lends support to the claim previously made for the Daniell estimator. Note that the Daniell estimate of PSD is a nonnegative function by its very definition, (2.7.20); such is not necessarily the case for several members of the Blackman–Tukey class of PSD estimators.

The lag window corresponding to the $W(\omega)$ in (2.7.22) is readily evaluated as follows:

$$\begin{aligned} w(k) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)e^{i\omega k}d\omega = \frac{1}{2\pi\beta}\int_{-\pi\beta}^{\pi\beta} e^{i\omega k}d\omega \\ &= \frac{\sin(k\pi\beta)}{k\pi\beta} = \text{sinc}(k\pi\beta) \end{aligned} \tag{2.7.23}$$

Note that $w(k)$ does not vanish as k increases; this effect leads to a subtle (but not essential) difference between the lag-windowed forms of the Daniell and Blackman–Tukey estimators. Since the inverse DTFT of $\hat{\phi}_p(\omega)$ is given by the sequence $\{\ldots, 0, 0, \hat{r}(-(N-1)), \ldots, \hat{r}(N-1), 0, 0, \ldots\}$, it follows immediately from (2.7.20) that $\hat{\phi}_D(\omega)$ can also be written as

$$\hat{\phi}_D(\omega) = \sum_{k=-(N-1)}^{N-1} w(k)\hat{r}(k)e^{-i\omega k} \tag{2.7.24}$$

It is seen from (2.7.24) that, like the Blackman–Tukey estimator, $\hat{\phi}_D(\omega)$ is a windowed version of the correlogram but, unlike the Blackman–Tukey estimator, the sum in (2.7.24) is not truncated to a value $M < N$. Hence, contrary to what might have been expected intuitively, the parameter

M defined in (2.7.21) cannot be exactly interpreted as a "truncation point" for the lag-windowed version of $\hat{\phi}_D(\omega)$. However, the equivalent bandwidth of $W(\omega)$ is clearly equal to β:

$$\beta_e = \beta$$

Thus, it follows that the equivalent time width of $w(k)$ is

$$N_e = 1/\beta_e = M$$

which shows that M plays essentially the same role here as the "truncation point" in the Blackman–Tukey estimator (and, indeed, it can be verified that $w(k)$ in (2.7.23) takes small values for $|k| > M$).

In closing this section and this chapter, we point out that the periodogram-based methods for spectrum estimation are all variations on the same theme. These methods attempt to reduce the variance of the basic periodogram estimator, at the expense of some reduction in resolution, by various means: averaging periodograms derived from data subsamples (Bartlett and Welch methods); averaging periodogram values locally around the frequency of interest (Daniell method); and smoothing the periodogram (Blackman–Tukey method). The unifying theme of these methods is seen in that they are essentially special forms of the Blackman–Tukey approach. In Chapter 5, we will push the unifying theme one step further, by showing that the periodogram-based methods can also be obtained as special cases of the filter-bank approach to spectrum estimation described there. (See also [Mullis and Scharf 1991].)

Finally, it is interesting to note that, although the modifications of the periodogram described in this chapter are indeed required when estimating a continuous PSD, the *unmodified periodogram* can be shown to be a satisfactory estimator (actually, the best one in large samples) for discrete (or line) spectra corresponding to sinusoidal signals. This is shown in Chapter 4.

2.8 COMPLEMENTS

2.8.1 Sample Covariance Computation via FFT

Computation of the sample covariances is a ubiquitous problem in spectral estimation and in signal processing applications. In this complement, we make use of the DTFT-like formula (2.2.2), relating the periodogram and the sample covariance sequence, to devise an FFT-based algorithm for computation of the $\{\hat{r}(k)\}_{k=0}^{N-1}$. We also compare the computational requirements of such an algorithm with those corresponding to the evaluation of $\{\hat{r}(k)\}$ via the temporal averaging formula (2.2.4), and we show that the former could be computationally more efficient than the latter if N is larger than a certain value.

From (2.2.2) and (2.2.6), we have (omitting the subscript p of $\hat{\phi}_p(\omega)$ for notational simplicity)

$$\hat{\phi}(\omega) = \sum_{k=-N+1}^{N-1} \hat{r}(k)e^{-i\omega k} = \sum_{p=1}^{2N-1} \hat{r}(p-N)e^{-i\omega(p-N)}$$

or, equivalently,

$$e^{-i\omega(N-1)}\ \hat{\phi}(\omega) = \sum_{p=1}^{2N-1} \rho(p)e^{-i\omega(p-1)} \tag{2.8.1}$$

where $\rho(p) \triangleq \hat{r}(p-N)$. Equation (2.8.1) has the standard form of a DFT (see (2.3.3)). It is evident from (2.8.1) that, in order to determine the sample covariance sequence, we need at least $(2N-1)$ values of the periodogram. This is expected; the sequence $\{\hat{r}(k)\}_{k=0}^{N-1}$ contains $(2N-1)$ real-valued unknowns, for the computation of which at least $(2N-1)$ periodogram values should be necessary (because $\hat{\phi}(\omega)$ is real valued).

Let

$$\omega_k = \frac{2\pi}{2N-1}\,(k-1), \qquad k = 1, \ldots, 2N-1$$

Also, let the sequence $\{y(t)\}_{t=1}^{2N-1}$ be obtained by padding the raw data sequence with $(N-1)$ zeroes. Compute

$$Y_k = \sum_{t=1}^{2N-1} y(t)e^{-i\omega_k(t-1)} \qquad (k = 1, 2, \ldots, 2N-1) \tag{2.8.2}$$

by means of a $(2N-1)$-point FFT algorithm. Next, evaluate

$$\tilde{\phi}_k = e^{-i\omega_k(N-1)}\ |Y_k|^2/N \qquad (k = 1, \ldots, 2N-1) \tag{2.8.3}$$

Finally, calculate the sample covariances via the "inversion" of (2.8.1):

$$\begin{aligned}\rho(p) &= \sum_{k=1}^{2N-1} \tilde{\phi}_k e^{i\omega_k(p-1)}/(2N-1)\\ &= \sum_{k=1}^{2N-1} \tilde{\phi}_k e^{i\omega_p(p-1)}/(2N-1)\end{aligned} \tag{2.8.4}$$

The previous computation may once again be done by using a $(2N-1)$-point FFT algorithm. The bulk of the procedure just outlined consists of the FFT-based computation of (2.8.2) and (2.8.4). That computation requires about $2N\log_2(2N)$ flops (assuming that the radix-2 FFT algorithm is used; the required number of operations is larger than the one previously given whenever N is not a power of two). The direct evaluation of the sample covariance sequence via (2.2.4) requires

$$N + (N-1) + \cdots + 1 \simeq N^2/2 \qquad \text{flops}$$

Hence, the FFT-based computation would be more efficient whenever

$$N > 4\log_2(2N)$$

This inequality is satisfied for $N \geq 32$. (Actually, N needs to be greater than 32, because we neglected the operations needed to implement equation (2.8.3).)

The previous discussion assumes that N is a power of two. If this is not the case, then the relative computational efficiency of the two procedures might be different. Note, also, that there are several other issues that could affect this comparison. For instance, if only the lags $\{\hat{r}(k)\}_{k=0}^{M-1}$ (with $M \ll N$) are required, then the number of computations required by (2.2.4) is drastically reduced. On the other hand, the FFT-based procedure can also be implemented in a more efficient way in such a case, and so it would remain more efficient computationally than a direct calculation when, for example, $N \geq 100$ [Oppenheim and Schafer 1989]. We conclude that the various implementation details could change the value of N beyond which the FFT-based procedure is more efficient than the direct approach and hence might influence the decision as to which of the two procedures should be used in a given application.

2.8.2 FFT-Based Computation of Windowed Blackman–Tukey Periodograms

The windowed Blackman–Tukey periodogram (2.5.1), unlike its unwindowed version, is not amenable to a direct computation via a single FFT. In this complement, we show that three FFTs are sufficient to evaluate (2.5.1): two FFTs for the computation of the sample covariance sequence entering the equation (2.5.1) (as described in Complement 2.8.1), and one FFT for the evaluation of (2.5.1). We also show that the computational formula for $\{\hat{r}(k)\}$ derived in Complement 2.8.1 can be used to obtain an FFT-based algorithm for evaluation of (2.5.1) directly in terms of $\hat{\phi}_p(\omega)$. We relate the latter way of computing (2.5.1) to the evaluation of $\hat{\phi}_{BT}(\omega)$ from the integral equation (2.5.3). Finally, we compare the two ways just outlined for evaluating the windowed Blackman–Tukey periodogram.

The windowed Blackman–Tukey periodogram can be written as

$$\begin{aligned}\hat{\phi}_{BT}(\omega) &= \sum_{k=-(N-1)}^{N-1} w(k)\hat{r}(k)e^{-i\omega k} \\ &= \sum_{k=0}^{N-1} w(k)\hat{r}(k)e^{-i\omega k} + \sum_{k=0}^{N-1} w(k)\hat{r}^*(k)e^{i\omega k} - w(0)\hat{r}(0) \\ &= 2\,\mathrm{Re}\left\{\sum_{k=0}^{N-1} w(k)\hat{r}(k)e^{-i\omega k}\right\} - w(0)\hat{r}(0) \qquad (2.8.5)\end{aligned}$$

where we made use of the facts that the window sequence is even and $\hat{r}(-k) = \hat{r}^*(k)$. It is now evident that an N-point FFT can be used to evaluate $\hat{\phi}_{BT}(\omega)$ at $\omega = 2\pi k/N$ $(k = 0, \ldots, N-1)$. This requires about $\frac{1}{2}N\log_2(N)$ flops that should be added to the $2N\log_2(2N)$ flops required to compute $\{\hat{r}(k)\}$ (as in Complement 2.8.1), giving a total of about $N[\frac{1}{2}\log_2(N) + 2\log_2(2N)]$ flops for this way of evaluating $\hat{\phi}_{BT}(\omega)$.

Next, we make use of the expression (2.8.4) for $\{\hat{r}(k)\}$ that is derived in Complement 2.8.1. We have

$$\hat{r}(p-N) = \frac{1}{2N-1}\sum_{k=1}^{2N-1}\hat{\phi}(\bar{\omega}_k)e^{i\bar{\omega}_k(p-N)} \qquad (p = 1, \ldots, 2N-1) \tag{2.8.6}$$

where $\bar{\omega}_k = 2\pi(k-1)/(2N-1)$ for $(k = 1, \ldots, 2N-1)$ and where $\hat{\phi}(\omega)$ is the unwindowed periodogram. Inserting (2.8.6) into (2.5.1), we obtain

$$\begin{aligned}\hat{\phi}_{BT}(\omega) &= \frac{1}{2N-1}\sum_{s=-(N-1)}^{N-1} w(s)e^{-i\omega s}\sum_{k=1}^{2N-1}\hat{\phi}(\bar{\omega}_k)e^{i\bar{\omega}_k s} \\ &= \frac{1}{2N-1}\sum_{k=1}^{2N-1}\hat{\phi}(\bar{\omega}_k)\left[\sum_{s=-(N-1)}^{N-1} w(s)e^{-i(\omega-\bar{\omega}_k)s}\right]\end{aligned} \tag{2.8.7}$$

which gives

$$\boxed{\hat{\phi}_{BT}(\omega) = \frac{1}{2N-1}\sum_{k=1}^{2N-1}\hat{\phi}(\bar{\omega}_k)\ W(\omega-\bar{\omega}_k)} \tag{2.8.8}$$

where $W(\omega)$ is the spectral window.

It might be thought that the last step in the preceding derivation requires that $\{w(k)\}$ be a "truncated-type" window (i.e., $w(k) = 0$ for $|k| \geq N$). However, no such requirement on $\{w(k)\}$ is needed: By inserting the usual expression for $\hat{\phi}(\omega)$ into (2.8.6), we obtain

$$\begin{aligned}\hat{r}(p-N) &= \frac{1}{2N-1}\sum_{k=1}^{2N-1}\left[\sum_{s=-(N-1)}^{N-1}\hat{r}(s)e^{-i\bar{\omega}_k s}\right]e^{i\bar{\omega}_k(p-N)} \\ &= \frac{1}{2N-1}\sum_{s=-(N-1)}^{N-1}\hat{r}(s)\left[\sum_{k=1}^{2N-1}e^{i\bar{\omega}_k(p-N-s)}\right] \\ &\triangleq \frac{1}{2N-1}\sum_{s=-(N-1)}^{N-1}\hat{r}(s)\Delta(s,p)\end{aligned}$$

where

$$\Delta(s,p) = \sum_{k=1}^{2N-1} e^{i\bar{\omega}_{p-N-s}k} = e^{i\bar{\omega}_{p-N-s}} \frac{e^{i(2N-1)\bar{\omega}_{p-N-s}} - 1}{e^{i\bar{\omega}_{p-N-s}} - 1}$$

Now, $(2N-1)\bar{\omega}_{p-N-s} = 2\pi(p-N-s)$, so it follows that

$$\Delta(s,p) = (2N-1)\delta_{p-N,s}$$

from which we immediately get

$$\frac{1}{2N-1} \sum_{s=-(N-1)}^{N-1} \hat{r}(s)\Delta(s,p) = \begin{cases} \hat{r}(p-N) & p = 1, \ldots, 2N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.8.9}$$

First, this calculation provides a cross-checking of the derivation of equation (2.8.6) in Complement 2.8.1. Second, the result (2.8.9) implies that the values of $\hat{r}(p-N)$ calculated with the formula (2.8.6) are equal to zero for $p < 1$ or $p > 2N-1$. It follows that the limits for the summation over s in (2.8.7) can be extended to $\pm\infty$, which shows that (2.8.8) is valid for an arbitrary window.

In the general case, there seems to be no way of evaluating (2.8.8) by means of an FFT algorithm. Hence, it appears that, for a general window, it is more efficient to base the computation of $\hat{\phi}_{BT}(\omega)$ on (2.8.5) rather than on (2.8.8). For certain windows, however, (2.8.8) could be more efficient than (2.8.5) computationally. For instance, in the case of the Daniell method, which corresponds to a rectangular spectral window, (2.8.8) takes a very convenient computational form and should be preferred to (2.8.5). It should be noted that (2.8.8) can be viewed as an *exact formula* for evaluation of the integral in equation (2.5.3). In particular, (2.8.8) provides an *exact* implementation formula for the Daniell periodogram (2.7.20) (whereas (2.7.16) is only an approximation of the integral (2.7.20) that is valid for sufficiently large values of N).

2.8.3 Data- and Frequency-Dependent Temporal Windows: The Apodization Approach

All windows discussed so far are both data- and frequency-independent; in other words, the window used is the same at any frequency of the spectrum and for any data sequence. Apparently, this is a rather serious restriction. A consequence of this restriction is that, for such nonadaptive windows (i.e., windows that do not adapt to the data under analysis) any attempt to reduce the leakage effect (by keeping the sidelobes low) inherently leads to a reduction of the resolution (due to the widening of the main lobe), and vice versa; see Section 2.6.1.

In this complement, we show how to design a *data- and frequency-dependent temporal window* that has the following desirable properties:

- It mitigates the leakage problem of the periodogram without compromising its resolution; and
- It does so with only a very marginal increase in the computational burden.

Our presentation is based on the *apodization approach* of [Stankwitz, Dallaire, and Fienup 1994], even though in some places we will deviate from it to some extent. Apodization is a term borrowed from optics where it has been used to mean a reduction of the sidelobes induced by diffraction.

We begin our presentation with a derivation of the temporally windowed periodogram, (2.6.24), in a least-squares (LS) framework. Consider the weighted LS fitting problem

$$\min_a \sum_{t=1}^{N} \rho(t) \left| y(t) - a e^{i\omega t} \right|^2 \tag{2.8.10}$$

where ω is given and so are the weights $\rho(t) \geq 0$. It can readily be verified that the minimizer of (2.8.10) is given by

$$\hat{a} = \frac{\sum_{t=1}^{N} \rho(t) y(t) e^{-i\omega t}}{\sum_{t=1}^{N} \rho(t)} \tag{2.8.11}$$

If we let

$$v(t) = \frac{\rho(t)}{\sum_{t=1}^{N} \rho(t)} \tag{2.8.12}$$

then we can rewrite (2.8.11) as a windowed DFT:

$$\hat{a} = \sum_{t=1}^{N} v(t) y(t) e^{-i\omega t} \tag{2.8.13}$$

The squared magnitude of (2.8.13) appears in the windowed periodogram formula (2.6.24)—of course, not accidentally, as $|\hat{a}|^2$ should indicate the power in $y(t)$ at frequency ω (*cf.* (2.8.10)).

The usefulness of the LS-based derivation of (2.6.24) lies in the fact that it reveals two constraints that must be satisfied by a temporal window, namely,

$$\boxed{v(t) \geq 0} \tag{2.8.14}$$

which follows from $\rho(t) \geq 0$, and

$$\boxed{\sum_{t=1}^{N} v(t) = 1} \tag{2.8.15}$$

which follows from (2.8.12). The constraint (2.8.15) can also be obtained by inspection of (2.6.24); indeed, if $y(t)$ had a component with frequency ω, then that component would pass undistorted (or unbiased) through the DFT in (2.6.24) if and only if (2.8.15) holds. For this reason, (2.8.15) is sometimes called the unbiasedness condition. On the other hand, the constraint (2.8.14) appears to be more difficult to obtain directly from (2.6.24).

Next, we turn our attention to window design, which is the problem of main interest here. To emphasize the dependence of the temporally windowed periodogram in (2.6.24) on $\{v(t)\}$, we use the notation $\hat{\phi}_v(\omega)$:

$$\hat{\phi}_v(\omega) = N \left| \sum_{t=1}^{N} v(t)y(t)e^{-i\omega t} \right|^2 \tag{2.8.16}$$

Note that, in (2.8.16), the squared modulus is multiplied by N, whereas, in (2.6.24), it is divided by N; this difference is due to the fact that the window $\{v(t)\}$ in this complement is constrained to satisfy (2.8.15), whereas in Section 2.6 it is implicitly assumed to satisfy $\sum_{t=1}^{N} v(t) = N$.

In the apodization approach, the window is selected such that

$$\boxed{\hat{\phi}_v(\omega) = \text{minimum}} \tag{2.8.17}$$

for *each* ω and for the given data sequence. Evidently, the apodization window will in general be both frequency and data dependent. Sometimes, such a window is said to be *frequency and data adaptive*. Let $\mathcal{C}$ denote the class of windows over which we perform the minimization in (2.8.17). Each window in $\mathcal{C}$ must satisfy the constraints (2.8.14) and (2.8.15). Usually, $\mathcal{C}$ is generated by an archetype window that depends on a number of unknown or free parameters, most commonly in a linear manner. It is important to observe that *we should not use more than two free parameters* to describe the windows $v(t) \in \mathcal{C}$. Indeed, one parameter is needed to satisfy the constraint (2.8.15) and the remaining one(s) to minimize the function in (2.8.17) under the inequality constraint (2.8.14); if, in the minimization operation, $\hat{\phi}_v(\omega)$ depends quadratically on more than one parameter, then in general the minimum value will be zero, $\hat{\phi}_v(\omega) = 0$ for all ω, which is not acceptable. We postpone a more detailed discussion on the parameterization of $\mathcal{C}$ until we have presented a motivation for the apodization design criterion in (2.8.17).

To understand *intuitively* why (2.8.17) makes sense, consider an example in which the data consists of two noise-free sinusoids. In this example, we use a rectangular window $\{v_1(t)\}$ and a Kaiser window $\{v_2(t)\}$. The use of these windows leads to the windowed periodograms in Figure 2.5. As is apparent from this figure, $v_1(t)$ is a "high-resolution" window that trades off leakage for resolution, whereas $v_2(t)$ compromises resolution (the two sinusoids are not resolved in the corresponding periodogram) for less leakage. By using the apodization principle in (2.8.17) to choose between $\hat{\phi}_{v_1}(\omega)$ and $\hat{\phi}_{v_2}(\omega)$, at each frequency ω, we obtain the spectral estimate shown in Figure 2.5, which inherits the high resolution of $\hat{\phi}_{v_1}(\omega)$ and the low leakage of $\hat{\phi}_{v_2}(\omega)$.

A *more formal* motivation of the apodization approach can be obtained as follows. Let

$$h_t = v(t)e^{-i\omega t}$$

In terms of $\{h_t\}$ the equality constraint (2.8.15) becomes

$$\sum_{t=1}^{N} h_t e^{i\omega t} = 1 \tag{2.8.18}$$

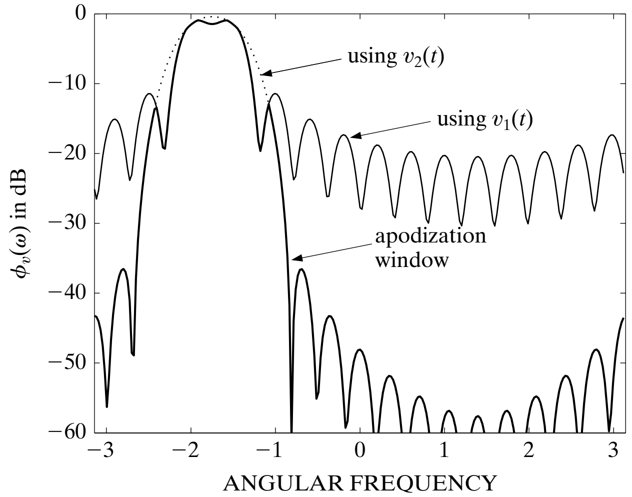


Figure 2.5 An apodization window design example using a rectangular window ($v_1(t)$) and a Kaiser window ($v_2(t)$). Shown are the periodograms corresponding to $v_1(t)$ and $v_2(t)$, and to the apodization window $v(t)$ selected using (2.8.17), for a data sequence of length 16 consisting of two noise-free sinusoids.

and hence the apodization design problem is to minimize

$$\left|\sum_{t=1}^{N} h_t y(t)\right|^2 \tag{2.8.19}$$

subject to (2.8.18), (2.8.14), and any other conditions resulting from the parameterization used for $\{v(t)\}$ (and therefore for $\{h_t\}$). We can interpret $\{h_t\}$ as an FIR filter of length N; consequently, (2.8.19) is the "power" of the filter output, and (2.8.18) is the (complex) gain of the filter at frequency ω. Therefore, by making use of $\{h_t\}$, we can describe the apodization principle in words as follows: Find the (parameterized) FIR filter $\{h_t\}$ that passes without distortion the sinusoid with frequency ω (see (2.8.18)) and minimizes the output power (see (2.8.19)) and thus attenuates any other frequency components in the data as much as possible. The (normalized) power at the output of the filter is taken as an estimate of the power in the data at frequency ω. This interpretation clearly can serve as a motivation of the apodization approach, and it sheds more light on the apodization principle. In effect, minimizing (2.8.19) subject to (2.8.18) (along with the other constraints on $\{h_t\}$ resulting from the parameterization used for $\{v(t)\}$) is a special case of a sound approach to spectral analysis that will be described in Section 5.4.1 (a fact noted, apparently for the first time, in [Lee and Munson Jr. 1995]).

As already stated previously, an important aspect that remains to be discussed is *the parameterization* of $\{v(t)\}$. For the apodization principle to make sense, the class $\mathcal{C}$ of windows must be chosen carefully. In particular, as just explained, we should not use more than two parameters to describe $\{v(t)\}$ (to prevent the meaningless "spectral estimate" $\hat{\phi}_v(\omega) \equiv 0$). The choice of the class $\mathcal{C}$ is also important from a *computational standpoint.* Indeed, the task of solving (2.8.17), for each ω and then computing the corresponding $\hat{\phi}_v(\omega)$ could be computationally demanding unless $\mathcal{C}$ is chosen carefully.

In what follows, we will consider the class of temporal windows used in [Stankwitz, Dallaire, and Fienup 1994]:

$$v(t) = \frac{1}{N}\left[\alpha - \beta \cos\left(\frac{2\pi}{N}t\right)\right], \quad t = 1, \ldots, N \tag{2.8.20}$$

It can readily be checked that (2.8.20) satisfies the constraints (2.8.14) and (2.8.15) if and only if

$$\alpha = 1 \text{ and } |\beta| \leq 1 \tag{2.8.21}$$

In addition, we require that

$$\beta \geq 0 \tag{2.8.22}$$

to ensure that the peak of $v(t)$ occurs in the middle of the interval $[1, N]$; this condition guarantees that the window in (2.8.20) (with $\beta > 0$) has lower sidelobes than the rectangular window corresponding to $\beta = 0$. (The window (2.8.20) with $\beta < 0$ generally has higher sidelobes than the rectangular window; hence, $\beta < 0$ cannot be a solution to the apodization design problem.)

Remark: The *temporal* window (2.8.20) is of the same type as the *lag* Hanning and Hamming windows in Table 2.1. For the latter windows, the interval of interest is $[-N, N]$; hence, for the peak of these windows to occur in the middle of the interval of interest, we need $\beta \leq 0$ (*cf.* Table 2.1). This observation explains the difference between (2.8.20) and the lag windows in Table 2.1. ■

Combining (2.8.20), (2.8.21), and (2.8.22) leads to the following (constrained) parameterization of the temporal windows:

$$\begin{aligned} v(t) &= \frac{1}{N}\left[1 - \beta \cos\left(\frac{2\pi}{N}t\right)\right] \\ &= \frac{1}{N}\left[1 - \frac{\beta}{2}\left(e^{i\frac{2\pi}{N}t} + e^{-i\frac{2\pi}{N}t}\right)\right], \quad \beta \in [0, 1] \end{aligned} \tag{2.8.23}$$

Assume, for simplicity, that N is a power of two (for the general case, we refer to [DeGraaf 1994]) and that a radix-2 FFT algorithm is used to compute

$$Y(k) = \sum_{t=1}^{N} y(t) e^{-i\frac{2\pi k}{N}t}, \quad k = 1, \ldots, N \tag{2.8.24}$$

as discussed in Section 2.3. Then the windowed periodogram corresponding to (2.8.23) can conveniently be computed, as follows:

$$\boxed{\hat{\phi}_v(k) = \frac{1}{N}\left|Y(k) - \frac{\beta}{2}\left[Y(k-1) + Y(k+1)\right]\right|^2, \quad k = 2, \ldots, N-1} \tag{2.8.25}$$

Furthermore, in (2.8.25), β is the solution to the following apodization design problem:

$$\min_{\beta \in [0,1]} \left|Y(k) - \frac{\beta}{2}\left[Y(k-1) + Y(k+1)\right]\right|^2 \tag{2.8.26}$$

The unconstrained minimizer of the preceding function is given by

$$\beta_0 = \text{Re}\left[\frac{2Y(k)}{Y(k-1) + Y(k+1)}\right] \tag{2.8.27}$$

Because the function in (2.8.26) is quadratic in β, it follows that the constrained minimizer of (2.8.26) is given by

$$\beta = \begin{cases} 0, & \text{if } \beta_0 < 0 \\ \beta_0, & \text{if } 0 \le \beta_0 \le 1 \\ 1, & \text{if } \beta_0 > 1 \end{cases} \tag{2.8.28}$$

Remark: It is interesting to note, from (2.8.28), that a change of the value of α in the window expression (2.8.20) will affect the apodization (optimal) window in a more complicated way than just a simple scaling. Indeed, if we change the value of α, for instance to $\alpha = 0.75$, then the interval for β becomes $\beta \in [0, 0.75]$ and this modification will affect the apodization window *nonlinearly* via (2.8.28). ■

The apodization-based windowed periodogram is obtained simply, by using the β given by (2.8.28) *in* (2.8.25). Hence, despite the fact that the apodization window is both frequency- and data-dependent (via β in (2.8.27), (2.8.28)), the implementation of the corresponding spectral estimate is only marginally more demanding computationally than is the implementation of an unwindowed periodogram. Compared with the latter, however, the apodization-based windowed periodogram has a considerably reduced leakage problem and essentially the same resolution. (See [Stankwitz, Dallaire, and Fienup 1994; DeGraaf 1994] for numerical examples illustrating this fact.)

2.8.4 Estimation of Cross-Spectra and Coherency Spectra

As can be seen from Complement 1.6.1, the estimation of the cross-spectrum $\phi_{yu}(\omega)$ of two stationary signals, $y(t)$ and $u(t)$, is a useful operation in the study of possible linear (dynamic) relations between $y(t)$ and $u(t)$. Let $z(t)$ denote the bivariate signal

$$z(t) = [y(t)\ u(t)]^T$$

and let

$$\hat{\phi}(\omega) = \frac{1}{N}\, Z(\omega) Z^*(\omega) \tag{2.8.29}$$

denote the unwindowed periodogram estimate of the spectral density matrix of $z(t)$. In equation (2.8.29),

$$Z(\omega) = \sum_{t=1}^{N} z(t) e^{-i\omega t}$$

is the DTFT of $\{z(t)\}_{t=1}^{N}$. Partition $\hat{\phi}(\omega)$ as

$$\hat{\phi}(\omega) = \begin{bmatrix} \hat{\phi}_{yy}(\omega) & \hat{\phi}_{yu}(\omega) \\ \hat{\phi}^*_{yu}(\omega) & \hat{\phi}_{uu}(\omega) \end{bmatrix} \tag{2.8.30}$$

As indicated by the notation previously used, estimates of $\phi_{yy}(\omega)$, of $\phi_{uu}(\omega)$, and of the cross-spectrum $\phi_{yu}(\omega)$ may be obtained from the corresponding elements of $\hat{\phi}(\omega)$.

We first show that the estimate of the coherence spectrum obtained from (2.8.30) is always such that

$$\left|\hat{C}_{yu}(\omega)\right| = 1 \qquad \text{for all } \omega \tag{2.8.31}$$

and, hence, it is useless. To see this, note that the rank of the 2 × 2 matrix in (2.8.30) is equal to one (see Result R22 in Appendix A), so we must have

$$\hat{\phi}_{uu}(\omega)\hat{\phi}_{yy}(\omega) = \left|\hat{\phi}_{yu}(\omega)\right|^2$$

which readily leads to the conclusion that the coherency spectrum estimate obtained from the elements of $\hat{\phi}(\omega)$ is bound to satisfy (2.8.31) and hence is meaningless. This result is yet another indication that the unwindowed periodogram is a poor estimate of the PSD.

Consider next a windowed Blackman–Tukey periodogram estimate of the cross-spectrum:

$$\boxed{\hat{\phi}_{yu}(\omega) = \sum_{k=-M}^{M} w(k)\hat{r}_{yu}(k) e^{-i\omega k}} \tag{2.8.32}$$

where $w(k)$ is the lag window, and $\hat{r}_{yu}(k)$ is some usual estimate of $r_{yu}(k)$. Unlike $r_{yy}(k)$ or $r_{uu}(k)$, $r_{yu}(k)$ does not necessarily peak at $k = 0$; moreover, it is in general not an even function.

The choice of the lag window for estimating cross-spectra may hence be governed by different rules from those commonly used in the autospectrum estimation.

The main task of a lag window is to retain the "essential part" of the covariance sequence in the defining equation for the spectral density. In this way, the bias is kept small and the variance is also reduced as the noisy tails of the sample covariance sequence are weighted out. For simplicity of discussion, assume that most of the area under the plot of $\hat{r}_{yu}(k)$ is concentrated about $k = k_0$, with $|k_0| \ll N$. As $\hat{r}_{yu}(k)$ is a reasonably accurate estimate of $r_{yu}(k)$, provided $|k| \ll N$, we can assume that $\{\hat{r}_{yu}(k)\}$ and $\{r_{yu}(k)\}$ have similar shapes. In such a case, one can redefine (2.8.32) as

$$\hat{\phi}_{yu}(\omega) = \sum_{k=-M}^{M} w(k - k_0)\hat{r}_{yu}(k)e^{-i\omega k}$$

where the lag window $w(s)$ is of the type recommended for autospectrum estimation. The choice of an appropriate value for k_0 in the preceding cross-spectral estimator is essential—if k_0 is selected poorly, the following situations can occur:

- If M is chosen small (to reduce the variance), the bias could be significant, because "essential" lags of the cross-covariance sequence could be left out.
- If M is chosen large (to reduce the bias), the variance could be significantly inflated when poorly estimated high-order "nonessential" lags are included in the spectral estimation formula.

Finally, let us look at the cross-spectrum estimators derived from (2.8.30) and (2.8.32), respectively, with a view to establishing a relation between them. Partition $Z(\omega)$ as

$$Z(\omega) = \begin{bmatrix} Y(\omega) \\ U(\omega) \end{bmatrix}$$

and observe that

$$\begin{aligned}
&\frac{1}{2\pi N} \int_{-\pi}^{\pi} Y(\omega)U^*(\omega)e^{i\omega k}\, d\omega \\
&= \frac{1}{2\pi N} \int_{-\pi}^{\pi} \sum_{t=1}^{N} \sum_{s=1}^{N} y(t)u^*(s)e^{-i\omega(t-s)}\; e^{i\omega k}\, d\omega \\
&= \frac{1}{N} \sum_{t=1}^{N} \sum_{s=1}^{N} y(t)u^*(s)\delta_{k,t-s} \\
&= \frac{1}{N} \sum_{t\in[1,N]\cap[1+k,N+k]} y(t)u^*(t-k) \triangleq \hat{r}_{yu}(k) \qquad (2.8.33)
\end{aligned}$$

where $\hat{r}_{yu}(k)$ can be rewritten in the following more familiar form:

$$\hat{r}_{yu}(k) = \begin{cases} \dfrac{1}{N} \displaystyle\sum_{t=k+1}^{N} y(t)u^*(t-k), & k = 0, 1, 2, \ldots \\ \dfrac{1}{N} \displaystyle\sum_{t=1}^{N+k} y(t)u^*(t-k), & k = 0, -1, -2, \ldots \end{cases}$$

Let

$$\hat{\phi}^p_{yu}(\omega) = \frac{1}{N} Y(\omega)U^*(\omega)$$

denote the unwindowed cross-spectral periodogram-like estimator, given by the off-diagonal element of $\hat{\phi}(\omega)$ in (2.8.30). With this notation, (2.8.33) can be written more compactly as

$$\hat{r}_{yu}(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{\phi}^p_{yu}(\mu)e^{i\mu k}\, d\mu$$

By using the previous equation in (2.8.32), we obtain

$$\begin{aligned} \hat{\phi}_{yu}(\omega) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{\phi}^p_{yu}(\mu) \sum_{k=-M}^{M} w(k)e^{-i(\omega-\mu)k}\, d\mu \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega-\mu)\hat{\phi}^p_{yu}(\mu)\, d\mu \end{aligned} \tag{2.8.34}$$

where $W(\omega) = \sum_{k=-\infty}^{\infty} w(k)e^{-i\omega k}$ is the spectral window. The previous equation should be compared with the similar equation, (2.5.3), that holds in the case of autospectra.

For implementation purposes, one can use the discrete approximation of (2.8.34) given that

$$\boxed{\hat{\phi}_{yu}(\omega) = \frac{1}{N} \sum_{k=-N}^{N} W(\omega-\omega_k)\hat{\phi}^p_{yu}(\omega_k)}$$

where $\omega_k = \frac{2\pi}{N}k$ are the Fourier frequencies. The periodogram (cross-spectral) estimate that appears in the above equation can be computed efficiently by means of an FFT algorithm.

2.8.5 More Time-Bandwidth Product Results

The time (or duration)-bandwidth product result (2.6.5) relies on the assumptions that both $w(t)$ and $W(\omega)$ have a dominant peak at the origin, that they both are real valued, and that they take on nonnegative values only. While most window-like signals (nearly) satisfy these assumptions, many other signals do not satisfy them. In this complement, we obtain time-bandwidth product results that apply to a much broader class of signals.

We begin by showing how the result (2.6.5) can be extended to a more general class of signals. Let $x(t)$ denote a general discrete-time sequence, and let $X(\omega)$ denote its DTFT. Both $x(t)$ and $X(\omega)$ are allowed to take negative or complex values, and neither is required to peak at the origin. Let t_0 and ω_0 denote the maximum points of $|x(t)|$ and $|X(\omega)|$, respectively. The time width (or duration) and bandwidth definitions in (2.6.1) and (2.6.2) are modified as follows:

$$\bar{N}_e = \frac{\sum_{t=-\infty}^{\infty} |x(t)|}{|x(t_0)|}$$

and

$$\bar{\beta}_e = \frac{\frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|\, d\omega}{|X(\omega_0)|}$$

Because $x(t)$ and $X(\omega)$ form a Fourier transform pair, we obtain

$$|X(\omega_0)| = \left| \sum_{t=-\infty}^{\infty} x(t)e^{-i\omega_0 t} \right| \le \sum_{t=-\infty}^{\infty} |x(t)|$$

and

$$|x(t_0)| = \left| \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{i\omega t_0}\, d\omega \right| \le \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|\, d\omega$$

which implies that

$$\boxed{\bar{N}_e \bar{\beta}_e \ge 1} \qquad (2.8.35)$$

This result, similar to (2.6.5), can be used to conclude the following:

$$\boxed{\text{A sequence } \{x(t)\} \text{ cannot be narrow in both time and frequency.}} \qquad (2.8.36)$$

More precisely, if $x(t)$ is narrow in one domain, it must be wide in the other domain. However, the inequality result (2.8.35), unlike (2.6.5), does not necessarily imply that $\bar{\beta}_e$ decreases whenever $\bar{N}_e$ increases (or vice versa). Furthermore, the result (2.8.35)—again unlike (2.6.5)—does not exclude the possibility that the signal is broad in both domains. In fact, in the general class of signals to which (2.8.35) applies, there are signals that are broad both in the time and in the frequency domain. (For such signals $\bar{N}_e \bar{\beta}_e \gg 1$; see, for example, [Papoulis 1977].) Evidently, the significant consequence of (2.8.35) is (2.8.36), which is precisely what makes the duration-bandwidth result an important one.

The duration-bandwidth product type of result (such as (2.6.5) or (2.8.35), and later (2.8.40)) has sometimes been referred to by using the generic name *uncertainty principle*, in an attempt to relate it to the Heisenberg Uncertainty Principle in quantum mechanics. (Briefly stated, the Heisenberg Uncertainty Principle asserts that the position and velocity of a particle cannot simultaneously be specified to arbitrary precision.) To support the relationship, one can argue as follows: Suppose that we are given a sequence with (equivalent) duration equal to N_e and that we are asked to use a linear filtering device to determine the sequence's spectral content in a certain narrow band. Because the filter impulse response cannot be longer than N_e (in fact, it should be (much) shorter!), it follows from the time-bandwidth product result that the filter's bandwidth can be on the order of $1/N_e$, *but not smaller*. Hence, the sequence's spectral content in fine bands on an order smaller than $1/N_e$ cannot be ascertained exactly and therefore is "uncertain." This is, in effect, the type of limitation that applies to the nonparametric spectral methods discussed in this chapter. However, this way of arguing is related to a *specific approach to spectral estimation and not to a fundamental limitation associated with the signal itself.* (As we will see in later chapters of this text, there are parametric methods of spectral analysis that can provide the "high resolution" necessary to determine the spectral content in bands that are on an order less than $1/N_e$.)

Next, we present another, slightly more general form of time-bandwidth product result. The definitions of duration and bandwidth used to obtain (2.8.35) make full sense whenever $|x(t)|$ and $|X(\omega)|$ are single pulse-like waveforms, though these definitions might give reasonable results in many other instances as well. There are several other possible definitions of the broadness of a waveform in either the time or the frequency domain. The definition used next and the corresponding time-bandwidth product result appear to be among the most general.

Let

$$\tilde{x}(t) = \frac{x(t)}{\sqrt{\sum_{t=-\infty}^{\infty} |x(t)|^2}} \tag{2.8.37}$$

and

$$\tilde{X}(\omega) = \frac{X(\omega)}{\sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega}} \tag{2.8.38}$$

By Parseval's theorem (see (1.2.6)), the denominators in (2.8.37) and (2.8.38) are equal to each other. Therefore, $\tilde{X}(\omega)$ is the DTFT of $\tilde{x}(t)$, as is already indicated by notation. Observe that

$$\sum_{t=-\infty}^{\infty} |\tilde{x}(t)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\tilde{X}(\omega)|^2 d\omega = 1$$

Hence, both $\{|\tilde{x}(t)|^2\}$ and $\{|\tilde{X}(\omega)|^2/2\pi\}$ can be interpreted as probability density functions in the sense that they are nonnegative and that they sum or integrate to one. The means and variances associated with these two "probability" densities are given by the following equations:

Time Domain:

$$\mu = \sum_{t=-\infty}^{\infty} t|\tilde{x}(t)|^2$$

$$\sigma^2 = \sum_{t=-\infty}^{\infty} (t-\mu)^2|\tilde{x}(t)|^2$$

Frequency Domain:

$$\nu = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi} \omega|\tilde{X}(\omega)|^2 d\omega$$
$$\rho^2 = \frac{1}{(2\pi)^3}\int_{-\pi}^{\pi} (\omega - 2\pi\nu)^2|\tilde{X}(\omega)|^2 d\omega$$

The values of the "standard deviations" σ and ρ show whether the normalized functions $\{|\tilde{x}(t)|\}$ and $\{|\tilde{X}(\omega)|\}$, respectively, are narrow or broad. Hence, *we can use σ and ρ as definitions for the duration and bandwidth, respectively,* of the original functions $\{x(t)\}$ and $\{X(\omega)\}$.

In what follows, we assume that

$$\mu = 0, \qquad \nu = 0 \tag{2.8.39}$$

For continuous-time signals, the zero-mean assumptions can always be made to hold by translating the origin appropriately on the time and frequency axes; see, for example, [Cohen 1995]. However, doing the same in the case of the discrete-time sequences considered here does not appear to be possible. Indeed, μ might not be integer valued, and the support of $X(\omega)$ is finite and, hence, affected by translation. Consequently, in the present case, the zero-mean assumption introduces some restriction; nevertheless, we impose it to simplify the analysis.

According to this discussion and assumption (2.8.39), we define the (equivalent) time width and bandwidth of $x(t)$ as follows:

$$\tilde{N}_e = \left[\sum_{t=-\infty}^{\infty} t^2|\tilde{x}(t)|^2\right]^{1/2}$$
$$\tilde{\beta}_e = \frac{1}{2\pi}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \omega^2|\tilde{X}(\omega)|^2 d\omega\right]^{1/2}$$

In the remainder of this complement, we prove the following time–bandwidth-product result:

$$\boxed{\tilde{N}_e\tilde{\beta}_e \geq \frac{1}{4\pi}} \tag{2.8.40}$$

which holds true under (2.8.39) and the weak additional assumption that

$$|\tilde{X}(\pi)| = 0 \tag{2.8.41}$$

To prove (2.8.40), first we note that

$$\tilde{X}'(\omega) \triangleq \frac{d\tilde{X}(\omega)}{d\omega} = -i\sum_{t=-\infty}^{\infty} t\tilde{x}(t)e^{-i\omega t}$$

Hence, $i\tilde{X}'(\omega)$ is the DTFT of $\{t\tilde{x}(t)\}$, which implies (by Parseval's theorem) that

$$\sum_{t=-\infty}^{\infty} t^2|\tilde{x}(t)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\tilde{X}'(\omega)|^2 d\omega \tag{2.8.42}$$

Consequently, by the Cauchy–Schwartz inequality for functions (see Result R23 in Appendix A),

$$\begin{aligned}
\tilde{N}_e\tilde{\beta}_e &= \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |\tilde{X}'(\omega)|^2 d\,\omega\right]^{1/2}\left[\frac{1}{(2\pi)^3}\int_{-\pi}^{\pi} \omega^2|\tilde{X}(\omega)|^2 d\omega\right]^{1/2} \\
&\geq \frac{1}{(2\pi)^2}\left|\int_{-\pi}^{\pi} \omega\tilde{X}^*(\omega)\tilde{X}'(\omega)d\omega\right| \\
&= \frac{1}{2(2\pi)^2}\left\{\left|\int_{-\pi}^{\pi} \omega\tilde{X}^*(\omega)\tilde{X}'(\omega)d\omega\right|\right. \\
&\qquad\qquad \left. + \left|\int_{-\pi}^{\pi} \omega\tilde{X}(\omega)\tilde{X}^{*\prime}(\omega)d\omega\right|\right\}
\end{aligned} \tag{2.8.43}$$

(The first equality above follows from (2.8.42) and the last one from a simple calculation.) Hence,

$$\begin{aligned}
\tilde{N}_e\tilde{\beta}_e &\geq \frac{1}{2(2\pi)^2}\left|\int_{-\pi}^{\pi} \omega\left[\tilde{X}^*(\omega)\tilde{X}'(\omega) + \tilde{X}(\omega)\tilde{X}^{*\prime}(\omega)\right]d\omega\right| \\
&= \frac{1}{2(2\pi)^2}\left|\int_{-\pi}^{\pi} \omega\left[|\tilde{X}(\omega)|^2\right]' d\omega\right|
\end{aligned}$$

which, after integration by parts and use of (2.8.41), yields

$$\tilde{N}_e\tilde{\beta}_e \geq \frac{1}{2(2\pi)^2}\left|\omega|\tilde{X}(\omega)|^2\Big|_{-\pi}^{\pi} - \int_{-\pi}^{\pi} |\tilde{X}(\omega)|^2 d\omega\right| = \frac{1}{2(2\pi)}$$

and the proof is concluded.

Remark: There is an alternative way to complete the proof above, starting from the inequality in (2.8.43). In fact, as we will see, this alternative proof yields a tighter inequality than (2.8.40). Let $\varphi(\omega)$ denote the phase of $\tilde{X}(\omega)$:

$$\tilde{X}(\omega) = |\tilde{X}(\omega)|e^{i\varphi(\omega)}$$

Then,

$$\begin{aligned}
\omega\tilde{X}^*(\omega)\tilde{X}'(\omega) &= \omega|\tilde{X}(\omega)|\left[|\tilde{X}(\omega)|\right]' + i\omega\varphi'(\omega)|\tilde{X}(\omega)|^2 \\
&= \frac{1}{2}\left[\omega|\tilde{X}(\omega)|^2\right]' - \frac{1}{2}|\tilde{X}(\omega)|^2 + i\omega\varphi'(\omega)|\tilde{X}(\omega)|^2
\end{aligned} \tag{2.8.44}$$

Inserting (2.8.44) into (2.8.43) yields

$$\tilde{N}_e \tilde{\beta}_e \geq \frac{1}{(2\pi)^2} \left| \frac{\omega}{2} |\tilde{X}(\omega)|^2 \Big|_{-\pi}^{\pi} - \pi + i2\pi\gamma \right| \tag{2.8.45}$$

where

$$\gamma = \frac{1}{2\pi} \int_{-\pi}^{\pi} \omega \varphi'(\omega) |\tilde{X}(\omega)|^2 d\omega$$

can be interpreted as the "covariance" of ω and $\varphi'(\omega)$ under the "probability density function" given by $|\tilde{X}(\omega)|^2/(2\pi)$. From (2.8.45), we at once obtain

$$\boxed{\tilde{N}_e \tilde{\beta}_e \geq \frac{1}{4\pi}\sqrt{1+4\gamma^2}} \tag{2.8.46}$$

which is a slightly stronger result than (2.8.40). ■

The results (2.8.40) and (2.8.46) are similar to (2.8.35); hence, the type of comment previously made about (2.8.35) applies to (2.8.40) and (2.8.46) as well.

For a time-bandwidth product result more general than (2.8.46), see [Doroslovacki 1998]; the papers [Calvez and Vilbé 1992] and [Ishii and Furukawa 1986] contain results similar to the one presented in this complement.

2.9 EXERCISES

Exercise 2.1: Covariance Estimation for Signals with Unknown Means

The sample-covariance estimators (2.2.3) and (2.2.4) are based on the assumption that the signal mean is equal to zero. A simple calculation shows that, under the zero-mean assumption,

$$E\{\tilde{r}(k)\} = r(k) \tag{2.9.1}$$

and

$$E\left\{\hat{r}(k)\right\} = \frac{N-|k|}{N}\, r(k) \tag{2.9.2}$$

where $\{\tilde{r}(k)\}$ denotes the sample covariance estimate in (2.2.3). Equations (2.9.1) and (2.9.2) show that $\tilde{r}(k)$ is an unbiased estimate of $r(k)$, whereas $\hat{r}(k)$ is a biased one. (Note, however, that the bias in $\hat{r}(k)$ is small for $N \gg |k|$.) For this reason, $\{\tilde{r}(k)\}$ and $\{\hat{r}(k)\}$ are often called, respectively, the unbiased and the biased sample covariances.

Whenever the signal mean is unknown, a most natural modification of the covariance estimators (2.2.3) and (2.2.4) is as follows:

$$\boxed{\tilde{r}(k) = \frac{1}{N-k} \sum_{t=k+1}^{N} [y(t) - \bar{y}]\, [y(t-k) - \bar{y}]^*} \tag{2.9.3}$$

and

$$\boxed{\hat{r}(k) = \frac{1}{N} \sum_{t=k+1}^{N} [y(t) - \bar{y}]\,[y(t-k) - \bar{y}]^*} \tag{2.9.4}$$

where $\bar{y}$ is the sample mean

$$\bar{y} = \frac{1}{N} \sum_{t=1}^{N} y(t) \tag{2.9.5}$$

Show that, in the unknown-mean case, the usual names, "unbiased" and "biased" sample covariance, associated with (2.9.3) and (2.9.4), respectively, might no longer be appropriate. Indeed, in such a case *both estimators could be biased*; furthermore, $\hat{r}(k)$ *could be less biased than* $\tilde{r}(k)$. To simplify the calculations, assume that $y(t)$ is white noise.

Exercise 2.2: Covariance Estimation for Signals with Unknown Means (cont'd)
Show that the sample covariance sequence $\{\hat{r}(k)\}$ in equation (2.9.4) of Exercise 2.1 satisfies the following equality:

$$\boxed{\sum_{k=-(N-1)}^{N-1} \hat{r}(k) = 0} \tag{2.9.6}$$

This equality might seem somewhat surprising. (Why should the $\{\hat{r}(k)\}$ satisfy such a constraint, which the true covariances do not necessarily satisfy? Note, for instance, that the latter covariance sequence could well comprise only positive elements.) However, the equality in (2.9.6) has a natural explanation when viewed in the context of periodogram-based spectral estimation. Derive and explain formula (2.9.6) in the aforementioned context.

Exercise 2.3: Unbiased ACS Estimates Can Lead to Negative Spectral Estimates
We stated in Section 2.2.2 that, if unbiased ACS estimates, given by equation (2.2.3), are used in the correlogram spectral estimate (2.2.2), then negative spectral estimates could result. Find an example data sequence $\{y(t)\}_{t=1}^{N}$ that gives such a negative spectral estimate.

Exercise 2.4: Variance of Estimated ACS
Let $\{y(t)\}_{t=1}^{N}$ be real Gaussian (for simplicity), with zero mean, ACS equal to $\{r(k)\}$, and ACS estimate (either biased or unbiased) equal to $\{\hat{r}(k)\}$ (given by equation (2.2.3) or (2.2.4); we treat both cases simultaneously). Assume, without loss of generality, that $k \geq 0$.

(a) Make use of equation (2.4.24) to show that

$$\operatorname{var}\{\hat{r}(k)\} = \alpha^2(k) \sum_{m=-(N-k-1)}^{N-k-1} (N - k - |m|) \left[r^2(m) + r(m+k)r(m-k)\right]$$

where

$$\alpha(k) = \begin{cases} \dfrac{1}{N-k} & \text{for unbiased ACS estimates} \\ \dfrac{1}{N} & \text{for biased ACS estimates} \end{cases}$$

Hence, for large N, the standard deviation of the ACS estimate is $\mathcal{O}(1/\sqrt{N})$ under weak conditions on the true ACS $\{r(k)\}$.

(b) For the special case that $y(t)$ is white Gaussian noise, show that $\text{cov}\{\hat{r}(k), \hat{r}(l)\} = 0$ for $k \neq l$, and find a simple expression for $\text{var}\{\hat{r}(k)\}$.

Exercise 2.5: Another Proof of the Equality $\hat{\phi}_p(\omega) = \hat{\phi}_c(\omega)$

The proof of the result (2.2.6) in the text introduces an auxiliary random sequence and treats the original data sequence as deterministic (nonrandom). That proof relies on several results previously derived. A more direct proof of (2.2.6) can be found using only (2.2.1), (2.2.2), and (2.2.4). Find such a proof.

Exercise 2.6: A Compact Expression for the Sample ACS

Show that the expressions for the sample ACS given in the text (equations (2.2.3) or (2.2.4) for $k \geq 0$ and (2.2.5) for $k < 0$) can be rewritten by using a single formula as follows:

$$\hat{r}(k) = \rho \sum_{p=1}^{N} \sum_{s=1}^{N} y(p) y^*(s) \delta_{s,p-k}, \quad k = 0, \pm 1, \ldots, \pm(N-1) \tag{2.9.7}$$

where $\rho = \frac{1}{N}$ for (2.2.4) and $\rho = \frac{1}{N-|k|}$ for (2.2.3).

Exercise 2.7: Yet Another Proof of the Equality $\hat{\phi}_p(\omega) = \hat{\phi}_c(\omega)$

Use the compact expression for the sample ACS derived in Exercise 2.6 to obtain a very simple proof of (2.2.6).

Exercise 2.8: Linear Transformation Interpretation of the DFT

Let F be the $N \times N$ matrix whose (k, t)th element is given by W^{kt}, where W is as defined in (2.3.2). Then the DFT, (2.3.3), can be written as a linear transformation of the data vector $y \triangleq [y(1) \ldots y(N)]^T$,

$$Y \triangleq [Y(0) \ldots Y(N-1)]^T = Fy \tag{2.9.8}$$

Show that F is an orthogonal matrix that satisfies

$$\frac{1}{N} FF^* = I \tag{2.9.9}$$

and, as a result, that the *inverse transform* is

$$y = \frac{1}{N} F^* Y \tag{2.9.10}$$

Deduce from these results that the DFT is nothing but a representation of the data vector y via an orthogonal basis in $\mathbf{C}^n$ (the basis vectors are the columns of F^*). Also, deduce that, if the sequence $\{y(t)\}$ is periodic with a period equal to N, then the Fourier coefficient vector, Y, determines the whole sequence $\{y(t)\}_{t=1,2,\ldots}$; and that, in effect, the inverse transform (2.9.10) can be extended to include all samples $y(1), \ldots, y(N),\ y(N+1),\ y(N+2), \ldots$

Exercise 2.9: For White Noise, the Periodogram Is an Unbiased PSD Estimator
Let $y(t)$ be a zero-mean white noise with variance σ^2 and let

$$Y(\omega_k) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} y(t) e^{-i\omega_k t}\ ; \quad \omega_k = \frac{2\pi}{N}\, k \qquad (k = 0, \ldots, N-1)$$

denote its (normalized) DFT evaluated at the Fourier frequencies.

- **(a)** Derive the covariances

$$E\left\{Y(\omega_k) Y^*(\omega_r)\right\}, \qquad k, r = 0, \ldots, N-1$$

- **(b)** Use the result of the previous calculation to conclude that the periodogram $\hat{\phi}(\omega_k) = |Y(\omega_k)|^2$ is an unbiased estimator of the PSD of $y(t)$.
- **(c)** Explain whether the unbiasedness property holds for $\omega \neq \omega_k$ as well. Present an intuitive explanation for your finding.

Exercise 2.10: Shrinking the Periodogram
First, we introduce a simple general result on mean squared error (MSE) reduction by shrinking. Let $\hat{x}$ be some estimate of a true (and unknown) parameter x. Assume that $\hat{x}$ is unbiased (i.e., $E\{\hat{x}\} = x$), and let $\sigma_{\hat{x}}^2$ denote the MSE of $\hat{x}$:

$$\sigma_{\hat{x}}^2 = E\left\{(\hat{x} - x)^2\right\}$$

(Since $\hat{x}$ is unbiased, $\sigma_{\hat{x}}^2$ also equals the variance of $\hat{x}$.) For a fixed (nonrandom) ρ, let

$$\tilde{x} = \rho \hat{x}$$

be another estimate of x. The "shrinkage coefficient" ρ can be chosen so as to make the MSE of $\tilde{x}$ (much) smaller than $\sigma_{\hat{x}}^2$. (Note that $\tilde{x}$, for $\rho \neq 1$, is a biased estimate of x; hence $\tilde{x}$ trades off bias for variance.) More precisely, show that the MSE of $\tilde{x}$, $\sigma_{\tilde{x}}^2$, achieves its minimum value (with respect to ρ),

$$\sigma_{\tilde{x}_o}^2 = \rho_o\, \sigma_{\hat{x}}^2$$

for

$$\rho_o = \frac{x^2}{x^2 + \sigma_{\hat{x}}^2}$$

Next, consider the application of the previous result to the periodogram. As we explained in the chapter, the periodogram-based spectral estimate is asymptotically unbiased and has an asymptotic MSE equal to the squared PSD value:

$$E\left\{\hat{\phi}_p(\omega)\right\} \to \phi(\omega), \qquad E\left\{\left(\hat{\phi}_p(\omega) - \phi(\omega)\right)^2\right\} \to \phi^2(\omega) \qquad \text{as } N \to \infty$$

Show that the "optimally shrunk" periodogram estimate is

$$\tilde{\phi}(\omega) = \hat{\phi}_p(\omega)/2$$

and that the MSE of $\tilde{\phi}(\omega)$ is 1/2 the MSE of $\hat{\phi}_p(\omega)$.

Finally, comment on the general applicability of this extremely simple tool for MSE reduction.

Exercise 2.11: Asymptotic Maximum Likelihood Estimation of $\phi(\omega)$ from $\hat{\phi}_p(\omega)$

It follows from the calculations in Section 2.4 that, asymptotically in N, $\hat{\phi}_p(\omega)$ has mean $\phi(\omega)$ and variance $\phi^2(\omega)$. In this exercise, we assume that $\hat{\phi}_p(\omega)$ is (asymptotically) Gaussian distributed (which is *not* necessarily the case; however, the spectral estimator derived here under the Gaussian assumption may also be used when this assumption does not hold). Hence, the asymptotic probability density function of $\hat{\phi}_p(\omega)$ is (we omit both the index p and the dependence on ω to simplify the notation):

$$p_\phi(\hat{\phi}) = \frac{1}{\sqrt{2\pi\phi^2}} \exp\left[-\frac{(\hat{\phi} - \phi)^2}{2\phi^2}\right]$$

Show that the maximum likelihood estimate (MLE) of ϕ based on $\hat{\phi}$, which by definition is equal to the maximizer of $p_\phi(\hat{\phi})$ (see Appendices B and C for a short introduction of maximum likelihood estimation), is given by

$$\tilde{\phi} = \frac{\sqrt{5} - 1}{2}\hat{\phi}$$

Compare $\tilde{\phi}$ with the "optimally shrunk" estimate of ϕ derived in Exercise 2.10.

Exercise 2.12: Plotting the Spectral Estimates in dB

It has been shown in this chapter that the spectral estimate $\hat{\phi}(\omega)$, obtained via an improved periodogram method, is asymptotically unbiased with a variance of the form $\mu^2\phi^2(\omega)$, where μ is a constant that can be made (much) smaller than 1 by choosing the window appropriately. This fact implies that the *confidence interval* $\hat{\phi}(\omega) \pm \mu\phi(\omega)$, constructed around the estimated PSD, should include the true (and unknown) PSD with a large probability. Now, obtaining a confidence interval as just shown has a twofold drawback: First, $\phi(\omega)$ is unknown; secondly, the interval could have significantly different widths for different frequency values.

Show that plotting $\hat{\phi}(\omega)$ in decibels eliminates the previous drawbacks. More precisely, show that, when $\hat{\phi}(\omega)$ is expressed in dB, its asymptotic variance is $c^2\mu^2$ (with $c = 10\log_{10} e$) and hence, that the confidence interval for a log-scale plot has the same width (independent of $\phi(\omega)$) for all ω.

Exercise 2.13: Finite-Sample Variance/Covariance Analysis of the Periodogram

This exercise has two aims. *First*, it shows that, in the Gaussian case, the variance/covariance analysis of the periodogram can be done in an extremely simple manner (even *without* the assumption that the data comes from a linear process, as in (2.4.26)). *Secondly*, the exercise asks for a finite-sample analysis which, for some purposes, might be more useful than the asymptotic analysis presented in the text. Indeed, the asymptotic-analysis result (2.4.21) can be misleading if not interpreted with care. For instance, (2.4.21) says that, asymptotically (for $N \to \infty$), $\hat{\phi}(\omega_1)$ and $\hat{\phi}(\omega_2)$ are uncorrelated with one another, no matter how close ω_1 and ω_2 are. This cannot be true in finite samples, and so the following question naturally arises: For a given N, how close can ω_1 be to ω_2 such that $\hat{\phi}(\omega_1)$ and $\hat{\phi}(\omega_2)$ are (nearly) uncorrelated with each other? The finite-sample analysis of this exercise can provide an answer to such questions, whereas the asymptotic analysis cannot.

Let

$$a(\omega) = [e^{i\omega} \dots e^{iN\omega}]^T$$

$$y = [y(1) \dots y(N)]^T$$

Then the periodogram, (2.2.1), can be written as follows (omitting the subindex p of $\hat{\phi}_p(\omega)$ in this exercise):

$$\hat{\phi}(\omega) = |a^*(\omega)y|^2/N \tag{2.9.11}$$

Assume that $\{y(t)\}$ is a zero-mean, stationary, circular Gaussian process. The "circular Gaussianity" assumption (see, for example, Appendix B) allows us to write the fourth-order moments of $\{y(t)\}$ (see equation (2.4.24)) as

$$\begin{aligned} E\left\{y(t)y^*(s)y(u)y^*(v)\right\} &= E\left\{y(t)y^*(s)\right\}E\left\{y(u)y^*(v)\right\} \\ &\quad + E\left\{y(t)y^*(v)\right\}E\left\{y(u)y^*(s)\right\} \end{aligned} \tag{2.9.12}$$

Make use of (2.9.11) and (2.9.12) to show that

$$\boxed{\begin{aligned} \text{cov}\{\hat{\phi}(\mu),\ \hat{\phi}(\nu)\} &\triangleq E\left\{\left[\hat{\phi}(\mu) - E\{\hat{\phi}(\mu)\}\right]\left[\hat{\phi}(\nu) - E\{\hat{\phi}(\nu)\}\right]\right\} \\ &= |a^*(\mu)Ra(\nu)|^2/N^2 \end{aligned}} \tag{2.9.13}$$

where $R = E\{yy^*\}$. Deduce from (2.9.13) that

$$\operatorname{var}\{\hat{\phi}(\mu)\} = |a^*(\mu)Ra(\mu)|^2/N^2 \tag{2.9.14}$$

Use (2.9.14) to readily rederive the variance part of the asymptotic result (2.4.21). Next, use (2.9.14) to show that *the covariance between $\hat{\phi}(\mu)$ and $\hat{\phi}(\nu)$ is not significant if*

$$|\mu - \nu| > 4\pi/N$$

and also that it can be significant otherwise. **Hint:** To show this inequality, make use of the Carathéodory parameterization of a covariance matrix in Section 4.9.2.

Exercise 2.14: Data-Weighted ACS Estimate Interpretation of Bartlett and Welch Methods
Consider the Bartlett estimator, and assume $LM = N$.

(a) Show that the Bartlett spectral estimate can be written as

$$\hat{\phi}_B(\omega) = \sum_{k=-(M-1)}^{M-1} \tilde{r}(k)e^{-i\omega k}$$

where

$$\tilde{r}(k) = \sum_{t=k+1}^{N} \alpha(k,t)y(t)y^*(t-k), \qquad 0 \le k < M$$

for some $\alpha(k,t)$ to be derived. Note that this is nearly in the form of the Blackman–Tukey spectral estimator, with the exception that the “standard” biased ACS estimate that is used in the Blackman–Tukey estimator is replaced by the “generalized” ACS estimate $\tilde{r}(k)$.

(b) Make use of the derived expression for $\alpha(k,t)$ to conclude that the Bartlett estimator is inferior to the Blackman–Tukey estimator (especially for small N) because it fails to use all available lag products in forming ACS estimates.

(c) Find $\alpha(k,t)$ for the Welch method. What overlap values (K in equation (2.7.7)) give lag product usage similar to that in the Blackman–Tukey method?

Exercise 2.15: Approximate Formula for Bandwidth Calculation
Let $W(\omega)$ denote a general spectral window that has a peak at $\omega = 0$ and is symmetric about that point. In addition, assume that the peak of $W(\omega)$ is narrow (as it usually should be). Under these assumptions, make use of a Taylor series expansion to show that an approximate formula for calculating *the bandwidth B of the peak of $W(\omega)$* is the following:

$$\boxed{B \simeq 2\sqrt{|W(0)/W''(0)|}} \tag{2.9.15}$$

The spectral-peak bandwidth B is defined mathematically as follows: Let ω_1 and ω_2 denote the "half-power points," defined through

$$W(\omega_1) = W(\omega_2) = W(0)/2, \qquad \omega_1 < \omega_2$$

(hence, the ratio $10\log_{10}\big(W(0)/W(\omega_j)\big) \simeq 3$ dB for $j = 1, 2$; we use $10\log_{10}$ rather than $20\log_{10}$ because the spectral window is applied to a power quantity, $\phi(\omega)$). Then, since $W(\omega)$ is symmetric, $\omega_2 = -\omega_1$, and

$$B \triangleq \omega_2 - \omega_1 = 2\omega_2$$

As an application of (2.9.15), show that

$$B \simeq 0.78 \cdot 2\pi/N \text{ (in radians per sampling interval)}$$

or, equivalently, that

$$B \simeq 0.78/N \text{ (in cycles per sampling interval)}$$

for the Bartlett window (2.4.15).

Note that this formula remains approximate even as $N \to \infty$. Even though the half-power bandwidth of the window gets smaller as N increases (so that one would expect the Taylor series expansion to be more accurate), the curvature of the window at $\omega = 0$ increases without bound as N increases. For the Bartlett window, verify that $B \simeq 0.9 \cdot 2\pi/N$ for N large, which differs from the prediction in this exercise by about 16%.

Exercise 2.16: A Further Look at the Time-Bandwidth Product

We saw in Section 2.6.1 that the product between the equivalent time and frequency widths of a regular window equals unity. Use the formula (2.9.15) derived in Exercise 2.15 to show that the spectral-peak bandwidth B of a window $w(k)$ that is nonzero only for $|k| < N$ satisfies

$$\boxed{B \cdot N \geq 1/\pi \quad \text{(in cycles per sampling interval)}} \tag{2.9.16}$$

This once again illustrates the "time-bandwidth product" type of result. Note that (2.9.16) involves the *effective* window time length and spectral *peak width*, as opposed to (2.6.5), which is concerned with *equivalent* time and frequency widths.

Exercise 2.17: Bias Considerations in Blackman–Tukey Window Design

The discussion in this chapter treated the bias of a spectral estimator and its resolution as two interrelated properties. This exercise illustrates further the strong relationship between bias and resolution.

Consider $\hat{\phi}_{BT}(\omega)$ as in (2.5.1), and, without loss of generality, assume that $E\{\hat{r}(k)\} = r(k)$. (Generality is not lost because, if $E\{\hat{r}(k)\} = \alpha(k)r(k)$, then replacing $w(k)$ by $\alpha(k)w(k)$ and $\hat{r}(k)$

by $\hat{r}(k)/\alpha(k)$ results in an equivalent estimator with unbiased ACS estimates.) Find the weights $\{w(k)\}_{k=-M+1}^{M-1}$ that minimize the squared bias, as given by the error measure

$$\epsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\phi(\omega) - E\{\hat{\phi}_{BT}(\omega)\} \right]^2 d\omega \qquad (2.9.17)$$

In particular, show that the weight function that minimizes ϵ is the rectangular window. Recall that the rectangular window also has the narrowest main lobe and, hence, the best resolution.

Exercise 2.18: A Property of the Bartlett Window

Let the window length, M, be given. Then, in the general case, the rectangular window can be expected to yield the windowed spectral estimate with the most favorable bias properties, because the sample covariance lags $\{\hat{r}(k)\}_{k=-(M-1)}^{M-1}$, appearing in (2.5.1), are left unchanged by this window (as in Exercise 2.17). The rectangular window, however, has the drawback that it is not positive definite and hence could produce negative spectral estimates. The Bartlett window, on the other hand, is positive definite and therefore yields a spectral estimate that is positive for all frequencies. Show that the latter window is the positive definite window that is closest to the rectangular one, in the sense of minimizing the following criterion:

$$\min_{\{w(k)\}} \sum_{k=0}^{M-1} |1 - w(k)| \quad \text{subject to:} \qquad (2.9.18)$$

1) $w(k) \equiv 0$ for $|k| \geq M$
2) $\{w(k)\}_{k=-\infty}^{\infty}$ is a positive definite sequence
3) $w(0) = 1$

Conclude that the Bartlett window is the positive definite window that distorts the sample covariances $\{\hat{r}(k)\}_{k=-(M-1)}^{M-1}$ *least* in the windowed spectral estimate formula. **Hint:** Any positive definite real window $\{w(k)\}_{k=-(M-1)}^{M-1}$ can be written as

$$w(k) = \sum_{i=0}^{M-1} b_i \, b_{i+k} \qquad (b_i = 0 \ \text{ for } \ i \geq M) \qquad (2.9.19)$$

for some real-valued parameters $\{b_i\}_{i=0}^{M-1}$. Make use of the preceding parameterization of the set of positive definite windows to transform (2.9.18) into an optimization problem without constraints.

COMPUTER EXERCISES

Tools for Periodogram Spectral Estimation:

The text website `www.prenhall.com/stoica` contains the following MATLAB functions for use in computing periodogram-based spectral estimates. In each case, `y` is the input data vector, `L` controls the frequency-sample spacing of the output, and the output vector is `phi`$= \phi(\omega_k)$, where $\omega_k = \frac{2\pi k}{L}$. MATLAB functions that generate the Correlogram, Blackman–Tukey, Windowed Periodogram, Bartlett, Welch, and Daniell spectral estimates are as follows:

- `phi = correlogramse(y,L)`
  Implements the correlogram spectral estimate in equation (2.2.2).
- `phi = btse(y,w,L)`
  Implements the Blackman–Tukey spectral estimate in equation (2.5.1); `w` is the vector $[w(0), \ldots, w(M-1)]^T$.
- `phi = periodogramse(y,v,L)`
  Implements the windowed periodogram spectral estimate in equation (2.6.24); `v` is a vector of window function elements $[v(1), \ldots, v(N)]^T$ and should be the same size as `y`. If `v` is a vector of ones, this function implements the unwindowed periodogram spectral estimate in equation (2.2.1).
- `phi = bartlettse(y,M,L)`
  Implements the Bartlett spectral estimate in equations (2.7.2) and (2.7.3); `M` is the size of each subsequence, as in equation (2.7.2).
- `phi = welchse(y,v,K,L)`
  Implements the Welch spectral estimate in equation (2.7.8); `M` is the size of each subsequence, `v` is the window function $[v(1), \ldots, v(M)]^T$ applied to each subsequence, and `K` is the overlap parameter, as in equation (2.7.7).
- `phi = daniellse(y,J,Ntilde)`
  Implements the Daniell spectral estimate in equation (2.7.16); `J` and `Ntilde` correspond to J and $\tilde{N}$ there.

Exercise C2.19: Zero-Padding Effects on Periodogram Estimators

In this exercise, we study the effect that zero padding has on the periodogram.

Consider the sequence

$$y(t) = 10\sin(0.2 \cdot 2\pi t + \phi_1) + 5\sin((0.2 + 1/N)2\pi t + \phi_2) + e(t), \tag{2.9.20}$$

where $t = 0, \ldots, N-1$, and $e(t)$ is white Gaussian noise with variance 1. Let $N = 64$ and $\phi_1 = \phi_2 = 0$.

From the results in Chapter 4, we find the spectrum of $y(t)$ to be

$$\begin{aligned}\phi(\omega) = {} & 50\pi\,[\delta(\omega - 0.2 \cdot 2\pi) + \delta(\omega + 0.2 \cdot 2\pi)] \\ & +12.5\pi\left[\delta(\omega - (0.2 + 1/N) \cdot 2\pi) + \delta(\omega + (0.2 + 1/N) \cdot 2\pi)\right] + 1\end{aligned}$$

Plot the periodogram for the sequence $\{y(t)\}$ and for the sequence $\{y(t)\}$ zero padded with N, $3N$, $5N$, and $7N$ zeroes.

Explain the difference between the five periodograms. Why does the first periodogram not give a good description of the spectral content of the signal? Note that zero padding does not change the resolution of the estimator.

Exercise C2.20: Resolution and Leakage Properties of the Periodogram
We have seen from Section 2.4 that the expected value of the periodogram is the convolution of the true spectrum $\phi_y(\omega)$ with the Fourier transform of a Bartlett window, denoted $W_B(\omega)$. (See equation (2.4.15).) The shape and size of the $W_B(\omega)$ function determines the amount of *smearing* and *leakage* in the periodogram. Similarly, in Section 2.5, we introduced a windowed periodogram in (2.6.24) whose expected value is equal to the expected value of a corresponding Blackman–Tukey estimate with weights $w(k)$ given by (2.6.31). Window functions different from the rectangular window could be used in the periodogram estimate, giving rise to correspondingly different windows in the correlogram estimate. The choice of window affects the resolution and leakage properties of the periodogram (correlogram) spectral estimate.

Resolution Properties: The amount of smearing of the spectral estimate is determined by the width of the main lobe, and the amount of leakage is determined by the energy in the sidelobes. The amount of smearing is what limits the resolving power of the periodogram; it is studied empirically next.

We first study the resolution properties by considering a sequence made up of two sinusoids in noise, where the two sinusoidal frequencies are "close". Consider

$$y(t) = a_1 \sin(f_0 \cdot 2\pi t + \phi_1) + a_2 \sin((f_0 + \alpha/N)2\pi t + \phi_2) + e(t), \tag{2.9.21}$$

where $e(t)$ is real-valued Gaussian white noise having mean zero and variance σ^2. We choose $f_0 = 0.2$ and $N = 256$, but the results are nearly independent of f_0 and N.

(a) Find empirically the 3 dB width of the main lobe of $W_B(\omega)$ as a function of N, and verify equation (2.4.18). Also, compute the peak sidelobe height (in dB) as a function of N. Note that the sidelobe level of a window function generally is independent of N. Verify this by examining plots of the magnitude of $W_B(\omega)$ for several values of N; try both linear and dB scales in your plots.

(b) Set $\sigma^2 = 0$ (to eliminate the statistical variation in the periodogram, so that the bias properties can be isolated and studied). Set $a_1 = a_2 = 1$ and $\phi_1 = \phi_2 = 0$. Plot the (zero-padded) periodogram of $y(t)$ for various α and determine the resolution threshold (i.e., the minimum value of α for which the two frequency components can be resolved). How does this value of α compare with the predicted resolution in Section 2.4?

(c) Repeat part (b) for a Hamming-windowed correlogram estimate.

(d) For reasonably high signal-to-noise ratio (SNR) values and reasonably close signal amplitudes, the resolution thresholds in parts (b) and (c) are not very sensitive to variations in the signal amplitudes and frequency f_0. However, these thresholds are sensitive to the phases ϕ_1 and ϕ_2, especially if α is smaller than 1. Try two pairs (ϕ_1, ϕ_2) such that the two sinusoids are in phase and out of phase, respectively, at the center of the observation interval, and compare the resolution thresholds. Also, try different values of a_1, a_2, and σ^2 to verify that their values have relatively little effect on the resolution threshold.

Spectral Leakage: In this part, we analyze the effects of leakage on the periodogram estimate. Leakage properties can be seen clearly when one is trying to estimate two sinusoidal terms that are well separated, but have greatly differing amplitudes.

(a) Generate the sinusoidal sequence for $\alpha = 4$, $\sigma^2 = 0$, and $\phi_1 = \phi_2 = 0$. Set $a_1 = 1$, and vary a_2. (Choose $a_2 = 1$, 0.1, 0.01, and 0.001, for example.) Compute the periodogram (using a rectangular data window), and comment on the ability to identify the second sinusoidal term from the spectral estimate.

(b) Repeat part (a) for $\alpha = 12$. Does the amplitude threshold for identifiability of the second sinusoidal term change?

(c) Explain your results in parts (a) and (b) by looking at the amplitude of the Bartlett window's Fourier transform at frequencies corresponding to α/N for $\alpha = 4$ and $\alpha = 12$.

(d) The Bartlett window has the property (as do many other windows) that the leakage level depends on the distance between spectral components in the data, as seen in parts (a) and (b). For many practical applications, it could be known what dynamic range the sinusoidal components in the data may have, and it is thus desirable to use a data window with a constant sidelobe level that can be chosen by the user. The Chebyshev window is a good choice for these applications, because the user can select the (constant) sidelobe level in the window design. (See the MATLAB command `chebwin`.)

Assume we know that the maximum dynamic range of sinusoidal components is 60 dB. Design a Chebyshev window $v(t)$ and corresponding Blackman–Tukey window $w(k)$, using (2.6.31), so that the two sinusoidal components of the data can be resolved for this dynamic range by using (i) the Blackman–Tukey spectral estimator with window $w(k)$, and (ii) the windowed periodogram method with window $v(t)$. Plot the Fourier transform of the window and determine the spectral resolution of the window.

Test your window design by computing the Blackman–Tukey and windowed periodogram estimates for two sinusoids whose amplitudes differ by 50 dB in dynamic range and whose frequency separation is the minimum value you predicted. Compare the resolution results with your predictions. Explain why the smaller-amplitude sinusoid can be detected by using one of the methods but not the other.

Exercise C2.21: Bias and Variance Properties of the Periodogram Spectral Estimate

In this exercise, we verify the theoretical predictions about bias and variance properties of the periodogram spectral estimate. We use autoregressive moving average (ARMA) signals (see Chapter 3) as test signals.

Bias Properties—Resolution and Leakage: We consider a random process

$$y(t) = H(z)e(t)$$

generated by filtering white noise, where $e(t)$ is zero-mean Gaussian white noise with variance $\sigma^2 = 1$ and the filter $H(z)$ is given by

$$H(z) = \sum_{k=1}^{2} A_k \left[\frac{1 - z_k z^{-1}}{1 - p_k z^{-1}} + \frac{1 - z_k^* z^{-1}}{1 - p_k^* z^{-1}} \right] \tag{2.9.22}$$

with

$$\begin{array}{ll} p_1 = 0.99e^{i2\pi 0.3} & p_2 = 0.99e^{i2\pi(0.3+\alpha)} \\ z_1 = 0.95e^{i2\pi 0.3} & z_2 = 0.95e^{i2\pi(0.3+\alpha)} \end{array} \tag{2.9.23}$$

We first let $A_1 = A_2 = 1$ and $\alpha = 0.05$.

- **(a)** Plot the true spectrum $\phi(\omega)$. Using a sufficiently fine grid for ω so that approximation errors are small, plot the ACS, using an inverse FFT of $\phi(\omega)$.
- **(b)** For $N = 64$, plot the Fourier transform of the Bartlett window, and also plot the expected value of the periodogram estimate $\hat{\phi}_p(\omega)$, as given by equation (2.4.8). We see that, for this example and data length, the main lobe width of the Bartlett window is wider than the distance between the spectral peaks in $\phi(\omega)$. Discuss how this relatively wide main lobe width affects the resolution properties of the estimator.
- **(c)** Generate 50 realizations of $y(t)$, each of length $N = 64$ data points. You can generate the data by passing white noise through the filter $H(z)$ (see the MATLAB commands `dimpulse` and `filter`); be sure to discard a sufficient number of initial filter output points to effectively remove the transient part of the filter output. Compute the periodogram spectral estimates for each data sequence; plot 10 spectral estimates overlaid on a single plot. Also, plot the average of the 50 spectral estimates. Compare the average with the predicted expected value found in part (b).
- **(d)** The resolution of the spectral peaks in $\phi(\omega)$ will depend on their separation relative to the width of the Bartlett-window main lobe. Generate realizations of $y(t)$ for $N = 256$, and find the minimum value of α such that the spectral peaks can be resolved in the averaged spectral estimate. Compare your results with the predicted formula (2.4.18) for spectral resolution.
- **(e)** Leakage from the Bartlett window will affect the ability to identify peaks of different amplitudes. To illustrate this, generate realizations of $y(t)$ for $N = 64$, for both $\alpha = 4/N$ and $\alpha = 12/N$. For each value of α, set $A_1 = 1$, and vary A_2 to find the minimum amplitude for which the lower-amplitude peak can reliably be identified from the averaged spectral estimate. Compare this value with the Bartlett window sidelobe level for $\omega = 2\pi\alpha$ and for the two values of α. Does the window sidelobe level accurately reflect the amplitude separation required to identify the second peak?

Variance Properties: In this part, we will verify that the variance of the periodogram is almost independent of the data length and will compare the empirical variance with theoretical predictions. For this part, we consider a broadband signal $y(t)$ for which the Bartlett window smearing and leakage effects are small.

Consider the broadband ARMA process

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

with

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

- **(a)** Plot the true spectrum $\phi(\omega)$.

(b) Generate 50 Monte Carlo data realizations, using different noise sequences, and compute the corresponding 50 periodogram spectral estimates. Plot the sample mean, the sample mean plus one sample standard deviation, and sample mean minus one sample standard deviation spectral estimate curves. Do this for $N = 64$, 256, and 1024. Note that the variance does not decrease with N.

(c) Compare the sample variance with the variance predicted in equation (2.4.21). It may help to plot $\text{stdev}\{\hat{\phi}(\omega)\}/\phi(\omega)$ and determine to what degree this curve is approximately constant. Discuss your results.

Exercise C2.22: Refined Methods: Variance–Resolution Tradeoff

In this exercise, we apply the Blackman–Tukey and Welch estimators to both a narrowband and a broadband random process. We consider the same processes in Chapters 3 and 5, to facilitate comparison with the spectral estimation methods developed in those chapters.

Broadband ARMA Process: Generate realizations of the broadband autoregressive moving-average (ARMA) process

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

with

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

$$B_1(z) = 1 + 0.3544z^{-1} + 0.3508z^{-2} + 0.1736z^{-3} + 0.2401z^{-4}$$

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

(a) Generate 50 Monte Carlo data realizations, using different noise sequences, and compute the corresponding 50 spectral estimates by using the following methods:

- The Blackman–Tukey spectral estimate using the Bartlett window $w_B(t)$. Try both $M = N/4$ and $M = N/16$.
- The Welch spectral estimate using the rectangular window $w_R(t)$, and using both $M = N/4$ and $M = N/16$ and overlap parameter $K = M/2$.

Plot the sample mean, the sample mean plus one sample standard deviation, and sample mean minus one sample standard deviation spectral estimate curves. Compare with the periodogram results from Exercise C2.21 and with each other.

(b) Judging from the plots you have obtained, how has the variance decreased in the refined estimates? How does this variance decrease as compared with the theoretical expectations?

(c) As discussed in the text, the value of M should be chosen to compromise between low "smearing" and low variance. For the Blackman–Tukey estimate, experiment with different values of M and different window functions to find a "best design" (in your judgment), and plot the corresponding spectral estimates.

Narrowband ARMA Process: Generate realizations of the narrowband ARMA process

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

with

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

and $N = 256$.

Repeat the experiments and comparisons in the broadband example for the narrowband process.

Exercise C2.23: Periodogram-Based 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 periodogram-based estimation techniques (possibly after some preprocessing as in part (b)) to estimate the spectral content of these data. Try to answer the following questions:

(a) Are there sinusoidal components (or periodic structure) in the data? If so, how many components, and at what frequencies?

(b) Nonlinear transformations and linear or polynomial trend removal are often applied before spectral analysis of a time series. For the `lynx` data, compare your spectral analysis results from the original data with those from the data transformed first by taking the logarithm of each sample and then by subtracting the sample mean of this logarithmic data. Does the logarithmic transformation make the data more sinusoidal in nature?

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:

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

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 $\varphi(t) = [y(t-1), \dots, y(t-n)]^T$. We interpret $\hat{y}(t)$ as a *linear prediction* of $y(t)$ from the n previous samples $y(t-1), \dots, y(t-n)$, 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

$$
\begin{aligned}
f(\theta) &= \sum_{t=N_1}^{N_2} |e(t)|^2 = \sum_{t=N_1}^{N_2} \left| y(t) + \sum_{i=1}^{n} a_i y(t-i) \right|^2 \\
&= \left\| \begin{bmatrix} y(N_1) \\ y(N_1+1) \\ \vdots \\ y(N_2) \end{bmatrix} + \begin{bmatrix} y(N_1-1) & \cdots & y(N_1-n) \\ y(N_1) & \cdots & y(N_1+1-n) \\ \vdots & & \vdots \\ y(N_2-1) & \cdots & y(N_2-n) \end{bmatrix} \theta \right\|^2 \\
&\triangleq \|y + Y\theta\|^2
\end{aligned}
\qquad (3.4.11)
$$

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

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

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

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

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 $1 \leq t \leq N$, 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 $x = [x_1 \ldots x_n]^T$, we define

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

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

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

where

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

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

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:

$$\begin{aligned} \theta_1 &= -\rho_1/\rho_0 = k_1 && \text{[1 flop]} \\ \sigma_1^2 &= \rho_0 - |\rho_1|^2/\rho_0 && \text{[1 flop]} \end{aligned}$$

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

$$\begin{aligned} k_{n+1} &= -\frac{\rho_{n+1} + \tilde{r}_n^*\theta_n}{\sigma_n^2} && [n+1 \text{ flops}] \\ \sigma_{n+1}^2 &= \sigma_n^2(1 - |k_{n+1}|^2) && \text{[2 flops]} \\ \theta_{n+1} &= \begin{bmatrix} \theta_n \\ 0 \end{bmatrix} + k_{n+1}\begin{bmatrix} \tilde{\theta}_n \\ 1 \end{bmatrix} && [n \text{ flops}] \end{aligned}$$

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 $\Delta_1=[1\ \ 1]^T$. 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 $(\times)$	no. of $(+)$
Initialization:		
$\Delta_0 = 1,\ \beta_0 = \rho_0,\ \gamma_0 = \rho_1$	–	–
$\Delta_1 = [1\ \ 1]^T,\ \beta_1 = \rho_0 + \rho_1,\ \gamma_1 = \rho_1 + \rho_2$	–	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:

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

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,

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

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

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

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

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

Finally, the ARMA spectrum is estimated as follows:

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

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

$$\begin{aligned} \varphi^T(t) &= [y(t-1), \dots, y(t-n) | -e(t-1), \dots, -e(t-m)] \\ \theta &= [a_1, \dots, a_n | b_1, \dots, b_m]^T \end{aligned}$$

Writing (3.7.10) in matrix form for $t = L+1, \dots, N$ (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]):

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

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

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

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

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

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

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

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:

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

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 $y(t-1), \ldots, y(t-j+1)$ 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

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

Then

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

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 $[1 \quad \theta]^T/\sigma^2$. 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

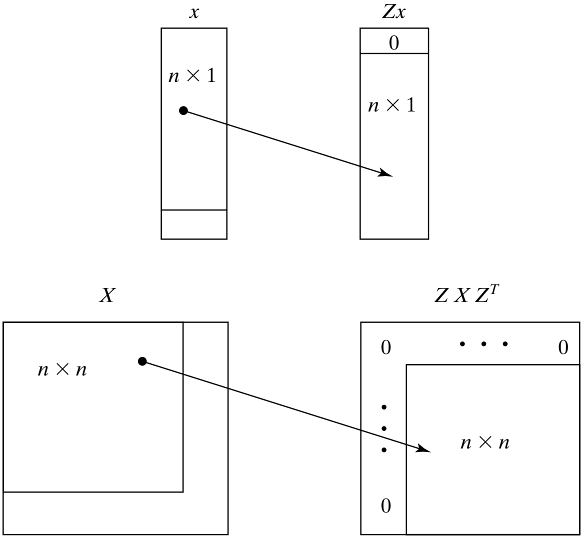
$$R_{n+1}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & R_n^{-1} \end{bmatrix} + \begin{bmatrix} 1 \\ \theta \end{bmatrix} [1 \quad \theta^*]/\sigma_n^2 \tag{3.9.17}$$

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

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

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

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

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

$$\vdots$$

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

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}:

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

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 $m \leq \hat{m}$. Indeed, if $m > \hat{m}$, we can extend the sequence $\{\hat{r}(k)\}$ with zeroes to make $m \leq \hat{m}$. 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

$$\hat{r} = \left[\ \hat{r}(0)\ \ldots\ \hat{r}(m)\ \right]^T$$

$$r = \left[\ r(0)\ \ldots\ r(m)\ \right]^T$$

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

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:

$$J_k = \begin{array}{c} \begin{array}{cccccc} \multicolumn{4}{c}{\overbrace{\hphantom{0\quad\dots\quad 0\quad 1}}^{k+1}} & & \end{array} \\ \left[\begin{array}{cccccc} 0 & \dots & 0 & 1 & & 0 \\ \vdots & & & 0 & \ddots & \\ & & & & \ddots & 1 \\ \vdots & & 0 & & & 0 \\ & & & & & \vdots \\ 0 & \dots & \dots & & \dots & 0 \end{array}\right] \end{array}, \quad (m+1) \times (m+1)$$

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

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

To prove this result, let

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

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 $k = 0, \ldots, m$)

$$\begin{aligned} r(k) &= \sigma^2 \sum_{j=k}^{m} b_j b_{j-k}^* = \sigma^2 \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix} J_k \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \\ &= \operatorname{tr} \left\{ J_k \cdot \sigma^2 \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix} \right\} \end{aligned} \tag{3.9.30}$$

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:

$$\alpha = \begin{bmatrix} Q_{1,1} \; Q_{1,2} \; \ldots \; Q_{1,m+1} \; ; \; Q_{2,2} \; \ldots \; Q_{2,m+1} \; ; \ldots \; ; Q_{m+1,m+1} \end{bmatrix}^T$$

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):

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

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)

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

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 $r(k) = \operatorname{tr}(J_k Q)$ 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

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

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 $\theta_b = [b_1 \dots b_n]^T$. 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 $k = 1, \dots, m$. 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

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

(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

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

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 $[\hat{\gamma}_0, \dots, \hat{\gamma}_m]^T$.
- `[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

$$\begin{aligned}\phi(\omega) = {} & 50\pi\,[\delta(\omega - 0.24\pi) + \delta(\omega + 0.24\pi)] \\ & + 12.5\pi\,[\delta(\omega - 0.26\pi) + \delta(\omega + 0.26\pi)] + \sigma^2\end{aligned}$$

(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$_\text{c}$, 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.

4

Parametric Methods for Line Spectra

4.1 INTRODUCTION

In several applications, particularly in communications, radar, sonar, and geophysical seismology, the signals dealt with can be described well by the *sinusoidal model*

$$\boxed{y(t) = x(t) + e(t) \quad ; \quad x(t) = \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \varphi_k)}} \tag{4.1.1}$$

where $x(t)$ denotes the noise-free complex-valued sinusoidal signal; $\{\alpha_k\}$, $\{\omega_k\}$, $\{\varphi_k\}$ are its *amplitudes*, *(angular) frequencies*, and *initial phases*, respectively; and $e(t)$ is an additive observation noise. The complex-valued form (4.1.1), of course, is not encountered in practice as it stands; practical signals are real valued. However, as already mentioned in Chapter 1, in many applications both the *in-phase and quadrature components* of the studied signal are available. (See Chapter 6 for more details on this aspect.) In the case of a (real-valued) sinusoidal signal, this means that both the sine and the corresponding cosine components are available. These two components may be processed by arranging them in a two-dimensional vector signal or a complex-valued signal of the form of (4.1.1). Since the complex-valued description (4.1.1) of the in-phase and quadrature components of a sinusoidal signal is the most convenient one from a mathematical standpoint, we focus on it in this chapter.

The noise $\{e(t)\}$ in (4.1.1) is usually assumed to be (complex-valued) *circular white noise*, defined in (2.4.19). We also make the white-noise assumption in this chapter. We may argue in the following way that the white-noise assumption is not particularly restrictive. Let the continuous-time counterpart of the noise in (4.1.1) be correlated, but assume that the "correlation time" of the continuous-time noise is less than half of the shortest period of the sine-wave components in the continuous-time counterpart of $x(t)$ in (4.1.1). If this mild condition is satisfied, then choosing the sampling period larger than the noise correlation time (yet smaller than half the shortest sinusoidal signal period, to avoid aliasing) results in a white discrete-time noise sequence $\{e(t)\}$. If the correlation condition above is not satisfied, but we know the shape of the noise spectrum, we can filter $y(t)$ by a linear *whitening filter* that makes the noise component at the filter output white; the sinusoidal components remain sinusoidal with the same frequencies, but with amplitudes and phases altered in a known way.

If the noise process is not white and has unknown spectral shape, then accurate frequency estimates can still be found if we estimate the sinusoids, using the nonlinear least squares (NLS) method in Section 4.3. (See [Stoica and Nehorai 1989b], for example.) Indeed, the properties of the NLS estimates in the colored-noise and unknown-noise cases are quite similar to those for the white-noise case, only with the sinusoidal signal amplitudes "adjusted" to give corresponding local SNRs—the signal-to-noise power ratio at each frequency ω_k. This amplitude adjustment is the same as that realized by the whitening filter approach. It is important to note that these comments apply only if the NLS method is used. The other estimation methods in this chapter (e.g., the subspace-based methods) depend on the assumption that the noise is white and can be affected adversely if the noise is not white (or is not prewhitened).

Concerning the signal in (4.1.1), we assume that $\omega_k \in [-\pi, \pi]$ and that $\alpha_k > 0$. We need to specify the sign of $\{\alpha_k\}$; otherwise we are left with a phase ambiguity. More precisely, without the condition $\alpha_k > 0$ in (4.1.1), both $\{\alpha_k, \omega_k, \varphi_k\}$ and $\{-\alpha_k, \omega_k, \varphi_k + \pi\}$ give the same signal $\{x(t)\}$, so the parameterization is not unique. As to the initial phases $\{\varphi_k\}$ in (4.1.1), one could assume that they are fixed (nonrandom) constants, which would result in $\{x(t)\}$ being a deterministic signal. In most applications, however, $\{\varphi_k\}$ are *nuisance parameters*, and it is more convenient to assume that they are random variables. Note that, if we try to mimic the conditions of a previous experiment as much as possible, we will usually be unable to ensure the same initial phases of the sine waves in the observed sinusoidal signal (this will be particularly true for received signals). Since there is usually no reason to believe that a specific set of initial phases is more likely than another one, or that two different initial phases are interrelated, we make the following assumption:

$$\boxed{\begin{array}{l}\text{The initial phases } \{\varphi_k\} \text{ are independent random variables uniformly distributed} \\ \text{on } [-\pi, \pi].\end{array}} \tag{4.1.2}$$

The covariance function and the PSD of the noisy sinusoidal signal $\{y(t)\}$ can be calculated in a straightforward manner under these assumptions. By using (4.1.2), we get

$$E\left\{e^{i\varphi_p} e^{-i\varphi_j}\right\} = 1 \quad \text{for} \quad p = j$$

and for $p \neq j$

$$\begin{aligned} E\left\{e^{i\varphi_p} e^{-i\varphi_j}\right\} &= E\left\{e^{i\varphi_p}\right\} E\left\{e^{-i\varphi_j}\right\} \\ &= \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{i\varphi} d\varphi\right]\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-i\varphi} d\varphi\right] = 0 \end{aligned}$$

Thus,

$$E\left\{e^{i\varphi_p} e^{-i\varphi_j}\right\} = \delta_{p,j} \tag{4.1.3}$$

Let

$$x_p(t) = \alpha_p e^{i(\omega_p t + \varphi_p)} \tag{4.1.4}$$

denote the pth sinusoid in (4.1.1). It follows from (4.1.3) that

$$E\left\{x_p(t) x_j^*(t-k)\right\} = \alpha_p^2 e^{i\omega_p k} \delta_{p,j} \tag{4.1.5}$$

which, in turn, gives

$$\boxed{r(k) = E\left\{y(t) y^*(t-k)\right\} = \sum_{p=1}^{n} \alpha_p^2 e^{i\omega_p k} + \sigma^2 \delta_{k,0}} \tag{4.1.6}$$

where σ^2 is the variance of $e(t)$. The derivation of the covariance function of $y(t)$ is completed. The PSD of $y(t)$ is given by the DTFT of $\{r(k)\}$ in (4.1.6), which is

$$\boxed{\phi(\omega) = 2\pi \sum_{p=1}^{n} \alpha_p^2 \delta(\omega - \omega_p) + \sigma^2} \tag{4.1.7}$$

where $\delta(\omega - \omega_p)$ is the Dirac impulse (or Dirac delta "function") which, by definition, has the property that

$$\int_{-\pi}^{\pi} F(\omega) \delta(\omega - \omega_p)\, d\omega = F(\omega_p) \tag{4.1.8}$$

for any function $F(\omega)$ that is continuous at ω_p. The expression (4.1.7) for $\phi(\omega)$ may be verified by inserting it in the inverse transform formula (1.3.8) and checking that the result is the covariance function. Doing so, we obtain

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[2\pi \sum_{p=1}^{n} \alpha_p^2 \delta(\omega - \omega_p) + \sigma^2\right] e^{i\omega k}\, d\omega = \sum_{p=1}^{n} \alpha_p^2 e^{i\omega_p k} + \sigma^2 \delta_{k,o} = r(k) \tag{4.1.9}$$

which is the desired result.

The PSD (4.1.7) is depicted in Figure 4.1. It consists of a “floor” of constant level equal to the noise power σ^2, along with n vertical lines (or impulses) located at the sinusoidal frequencies $\{\omega_k\}$ and having zero support but nonzero areas equal to 2π times the sine wave powers $\{\alpha_k^2\}$. Owing to its appearance, as exhibited in Figure 4.1, $\phi(\omega)$ in (4.1.7) is called a *line* or *discrete spectrum*.

It is evident from the previous discussion that a spectral analysis based on the parametric PSD model (4.1.7) reduces to the problem of estimating the parameters of the signal in (4.1.1). In most applications, such as those listed at the beginning of this chapter, the parameters of major interest are the locations of the spectral lines—namely, the sinusoidal frequencies. In the next sections, we present a number of methods for *spectral line analysis*. We focus on the problem of *frequency estimation*, meaning estimation of $\{\omega_k\}_{k=1}^n$ from a set of observations $\{y(t)\}_{t=1}^N$. Once the frequency estimates have been obtained, estimation of the other signal parameters (or PSD parameters) becomes a simple *linear regression problem*. More precisely, for given $\{\omega_k\}$, the observations $y(t)$ can be written as a linear regression function whose coefficients are equal to the remaining unknowns $\{\alpha_k e^{i\varphi_k} \triangleq \beta_k\}$:

$$y(t) = \sum_{k=1}^{n} \beta_k e^{i\omega_k t} + e(t) \tag{4.1.10}$$

If desired, $\{\beta_k\}$ (and hence $\{\alpha_k\}$, $\{\varphi_k\}$) in (4.1.10) can be obtained by a least-squares method (as in equation (4.3.8)). Alternatively, one may determine the signal powers $\{\alpha_k^2\}$—for given $\{\omega_k\}$—from the sample version of (4.1.6):

$$\hat{r}(k) = \sum_{p=1}^{n} \alpha_p^2 e^{i\omega_p k} + \text{residuals} \qquad \text{for } \ k \geq 1 \tag{4.1.11}$$

where the residuals arise from finite-sample estimation of $r(k)$; this is, once more, a linear regression with $\{\alpha_p^2\}$ as unknown coefficients. The solution to either linear regression problem is straightforward and is discussed in Section A.8 of Appendix A.

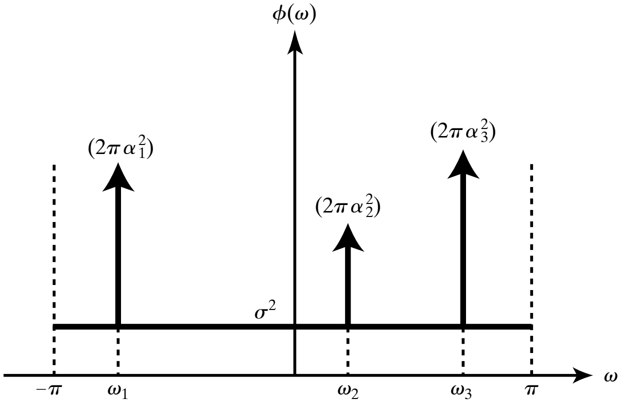


Figure 4.1 The PSD of a complex sinusoidal signal in additive white noise.

The methods for frequency estimation that will be described in the following sections are sometimes called *high-resolution* (or, even, *superresolution*) techniques. This is due to their ability to resolve spectral lines separated in frequency $f = \omega/2\pi$ by less than $1/N$ cycles per sampling interval, which is the resolution limit for the classical periodogram-based methods. All of the high-resolution methods to be discussed in the following provide *consistent estimates* of $\{\omega_k\}$ under the assumptions we made. Their consistency will surface in the following discussion in an obvious manner; hence, we do not need to pay special attention to this aspect. Nor do we discuss in detail other statistical properties of the frequency estimates obtained by these high-resolution methods, though in Appendix B we review the Cramér–Rao bound and the best accuracy that can be achieved by such methods. For derivations and discussions of the statistical properties not addressed in this text, we refer the interested reader to [Stoica, Söderström, and Ti 1989; Stoica and Söderström 1991; Stoica, Moses, Friedlander, and Söderström 1989; Stoica and Nehorai 1989b]. Let us briefly summarize the conclusions of these analyses: All the high-resolution methods presented in the following provide very accurate frequency estimates, with only small differences in their statistical performances. Furthermore, the computational burdens associated with these methods are rather similar. Hence, selecting one of the high-resolution methods for frequency estimation is essentially a "matter of taste," even though we will identify some advantages of one of these methods, named ESPRIT, over the others.

We should point out that the comparison in the previous paragraph between the high-resolution methods and the periodogram-based techniques is unfair, in the sense that periodogram-based methods do not assume any knowledge about the data, whereas high-resolution methods exploit an exact description of the studied signal. The additional information assumed allows a parametric method to offer better resolution than the nonparametric method of the periodogram. On the other hand, when no two spectral lines in the spectrum are separated by less than $1/N$, the *unmodified periodogram* turns out to be an excellent frequency estimator which may outperform any of the high-resolution methods (as we shall see). One may ask why the *unmodified* periodogram is preferred over the many windowed or smoothed periodogram techniques to which we paid so much attention in Chapter 2. The explanation actually follows from the discussion in that chapter. The unmodified periodogram can be viewed as a Blackman–Tukey "windowed" estimator with a rectangular window of maximum length equal to $2N + 1$. Of all window sequences, this is exactly the one which has the narrowest main lobe and, hence, the one that affords the maximum spectral resolution, a desirable property for high-resolution spectral-line scenarios. It should be noted, however, that if the sinusoidal components in the signal are not very closely spaced in frequency, but their amplitudes differ significantly from one another, then a mildly windowed periodogram (to avoid leakage) might perform better than the unwindowed periodogram. In the unwindowed periodogram, the weaker sinusoids could be obscured by the leakage from the stronger ones, and hence they might not be visible in a plot of the estimated spectrum.

In order to simplify the discussion in this chapter, we assume that the number of sinusoidal components, n, in (4.1.1) is known. When n is unknown, as could well be the case in many applications, it can be estimated from the available data, in a way, for example, described in [Fuchs 1988; Kay 1988; Marple 1987; Proakis, Rader, Ling, and Nikias 1992; Söderström and Stoica 1989] and in Appendix C.

4.2 MODELS OF SINUSOIDAL SIGNALS IN NOISE

The frequency estimation methods presented in this chapter rely on three different models for the noisy sinusoidal signal (4.1.1). This section introduces the three models of (4.1.1).

4.2.1 Nonlinear Regression Model

The nonlinear regression model is given by (4.1.1). Note that the $\{\omega_k\}$ enter in a nonlinear fashion in (4.1.1), hence the name "nonlinear regression" given to this type of model for $\{y(t)\}$. The other two models for $\{y(t)\}$, to be discussed in the following, are derived from (4.1.1); they are descriptions of the data that are not as complete as (4.1.1). However, they preserve the information required to extract the frequencies $\{\omega_k\}$ which, as already stated, are the parameters of major interest. Hence, in some sense, these two models are more appropriate for frequency estimation; they do not include some of the *nuisance parameters* that appear in (4.1.1).

4.2.2 ARMA Model

It can readily be verified that

$$(1 - e^{i\omega_k} z^{-1}) x_k(t) \equiv 0 \tag{4.2.1}$$

where z^{-1} denotes the unit delay (or shift) operator introduced in Chapter 1. Hence, $(1 - e^{i\omega_k} z^{-1})$ is an *annihilating filter* for the kth component in $x(t)$. By using this simple observation, we obtain the *homogeneous AR* equation for $\{x(t)\}$, namely,

$$\boxed{A(z)x(t) = 0} \tag{4.2.2}$$

and the *ARMA model* for the noisy data $\{y(t)\}$—that is,

$$\boxed{\begin{aligned} A(z)y(t) &= A(z)e(t) \\ A(z) &= \prod_{k=1}^{n} (1 - e^{i\omega_k} z^{-1}) \end{aligned}} \tag{4.2.3}$$

It may be a useful exercise to derive equation (4.2.2) in a different way. The PSD of $x(t)$ consists of n spectral lines located at $\{\omega_k\}_{k=1}^{n}$. It should then be clear, in view of the relation (1.4.9) governing the transfer of a PSD through a linear system, that any filter that has zeroes at frequencies $\{\omega_k\}$ is an annihilating filter for $x(t)$. The polynomial $A(z)$ in (4.2.3) is the simplest kind of such annihilating filter. This polynomial bears *complete information* about $\{\omega_k\}$; hence, the problem of estimating the frequencies can be reduced to that of determining $A(z)$.

We remark that the ARMA model (4.2.3) has a very special form (for which reason it is sometimes called a "degenerate" ARMA model). All its poles and zeroes are located exactly on the unit circle. Furthermore, its AR and MA parts are identical. It might be tempting to cancel the common poles and zeroes in (4.2.3). However, such an operation leads to the wrong conclusion that $y(t) = e(t)$ and, therefore, should be invalid. Let us explain briefly why cancelation in (4.2.3) is not allowed. The ARMA equation description of a signal $y(t)$ is *asymptotically* equivalent to the associated transfer-function description (in the sense that both give the same signal sequence, for $t \to \infty$) if and only if the poles are situated strictly inside the unit circle. If there are poles on the unit circle, then the equivalence between these two descriptions ceases. In particular, the solution of an ARMA equation with poles on the unit circle strongly depends on the initial conditions, whereas the transfer-function description does not impose a dependence on initial values.

4.2.3 Covariance Matrix Model

A notation that will often be used in what follows is

$$\boxed{\begin{aligned} a(\omega) &\triangleq [1 \;\; e^{-i\omega} \ldots e^{-i(m-1)\omega}]^T \qquad (m \times 1) \\ A &= [a(\omega_1) \ldots a(\omega_n)] \qquad (m \times n) \end{aligned}} \tag{4.2.4}$$

In (4.2.4), m is a positive integer not yet specified. The matrix A is a Vandermonde matrix, which enjoys the following rank property (see Result R24 in Appendix A):

$$\boxed{\operatorname{rank}(A) = n \quad \text{if} \;\; m \geq n \;\; \text{and} \;\; \omega_k \neq \omega_p \;\; \text{for} \;\; k \neq p} \tag{4.2.5}$$

By making use of the previous notation, along with (4.1.1) and (4.1.4), we can write

$$\boxed{\begin{aligned} \tilde{y}(t) &\triangleq \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-m+1) \end{bmatrix} = A\tilde{x}(t) + \tilde{e}(t) \\ \tilde{x}(t) &= [x_1(t) \ldots x_n(t)]^T \\ \tilde{e}(t) &= [e(t) \ldots e(t-m+1)]^T \end{aligned}} \tag{4.2.6}$$

The following expression for the covariance matrix of $\tilde{y}(t)$ can be readily derived from (4.1.5) and (4.2.6):

$$\boxed{R \triangleq E\left\{\tilde{y}(t)\tilde{y}^*(t)\right\} = APA^* + \sigma^2 I \quad ; \quad P = \begin{bmatrix} \alpha_1^2 & & 0 \\ & \ddots & \\ 0 & & \alpha_n^2 \end{bmatrix}} \tag{4.2.7}$$

This equation constitutes the covariance matrix model of the data. As we will show later, the *eigenstructure* of R contains complete information on the frequencies $\{\omega_k\}$, and this is exactly where the usefulness of (4.2.7) lies.

From equations (4.2.6) and (4.1.5), we also derive the following result for later use:

$$\boxed{\begin{aligned}\Gamma &\triangleq E\left\{\begin{bmatrix} y(t-L-1) \\ \vdots \\ y(t-L-M)\end{bmatrix}[y^*(t)\ldots y^*(t-L)]\right\} \\ &= E\left\{A_M\tilde{x}(t-L-1)\tilde{x}^*(t)A^*_{L+1}\right\} \\ &= A_M P_{L+1} A^*_{L+1} \qquad (L, M \geq 1)\end{aligned}} \tag{4.2.8}$$

Here A_K stands for A in (4.2.4) with $m = K$, and

$$P_K = \begin{bmatrix} \alpha_1^2 e^{-i\omega_1 K} & & 0 \\ & \ddots & \\ 0 & & \alpha_n^2 e^{-i\omega_n K}\end{bmatrix}$$

As we explain in detail later, the *null space* of the matrix Γ (with $L, M \geq n$) gives complete information on the frequencies $\{\omega_k\}$.

4.3 NONLINEAR LEAST-SQUARES METHOD

An intuitively appealing approach to spectral line analysis, based on the *nonlinear regression model* (4.1.1), consists of finding the unknown parameters as the minimizers of the criterion

$$f(\omega, \alpha, \varphi) = \sum_{t=1}^{N}\left| y(t) - \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \varphi_k)}\right|^2 \tag{4.3.1}$$

where ω is the vector of frequencies ω_k, and similarly for α and φ. The sinusoidal model determined as above has the smallest "sum of squares" distance to the observed data $\{y(t)\}_{t=1}^{N}$. Since f is a nonlinear function of its arguments $\{\omega, \varphi, \alpha\}$, the method that obtains parameter estimates by minimizing (4.3.1) is called the *nonlinear least-squares (NLS) method.* When the (white) noise $e(t)$ is Gaussian distributed, the minimization of (4.3.1) can also be interpreted as the *method of maximum likelihood* (see Appendices B and C); in that case, minimization of (4.3.1) can be shown to provide the parameter values most likely to "explain" the observed data sequence. (See [Söderström and Stoica 1989; Kay 1988; Marple 1987].)

The criterion in (4.3.1) depends on $\{\alpha_k\}$, $\{\varphi_k\}$, and $\{\omega_k\}$. However, it can be *concentrated with respect to the nuisance parameters* $\{\alpha_k, \varphi_k\}$, as explained next. By making use of the notation,

$$\beta_k = \alpha_k e^{i\varphi_k} \tag{4.3.2}$$

$$\beta = [\beta_1 \dots \beta_n]^T \tag{4.3.3}$$

$$Y = [y(1) \dots y(N)]^T \tag{4.3.4}$$

$$B = \begin{bmatrix} e^{i\omega_1} & \dots & e^{i\omega_n} \\ \vdots & & \vdots \\ e^{iN\omega_1} & \dots & e^{iN\omega_n} \end{bmatrix} \tag{4.3.5}$$

we can write the function f in (4.3.1) as

$$f = (Y - B\beta)^*(Y - B\beta) \tag{4.3.6}$$

The Vandermonde matrix B in (4.3.5) (which resembles the matrix A defined in (4.2.4)) has full column rank equal to n under the weak condition that $N \geq n$; in this case, $(B^*B)^{-1}$ exists. By using this observation, we can put (4.3.6) in the more convenient form:

$$\begin{aligned} f = [\beta - (B^*B)^{-1}B^*Y]^*[B^*B][\beta - (B^*B)^{-1}B^*Y] \\ + Y^*Y - Y^*B(B^*B)^{-1}B^*Y \end{aligned} \tag{4.3.7}$$

For any choice of $\omega = [\omega_1, \dots, \omega_n]^T$ in B (which is such that $\omega_k \neq \omega_p$ for $k \neq p$), we can choose β to make the first term of f zero; thus, we see that the vectors β and ω that minimize f are given by

$$\boxed{\begin{aligned} \hat{\omega} &= \arg\max_\omega [Y^*B(B^*B)^{-1}B^*Y] \\ \hat{\beta} &= (B^*B)^{-1}B^*Y|_{\omega=\hat{\omega}} \end{aligned}} \tag{4.3.8}$$

It can be shown that, as N tends to infinity, $\hat{\omega}$ obtained as in the preceding discussion converges to ω (i.e., $\hat{\omega}$ is a consistent estimate) and, in addition, the estimation errors $\{\hat{\omega}_k - \omega_k\}$ have the following (asymptotic) covariance matrix:

$$\boxed{\mathrm{Cov}(\hat{\omega}) = \frac{6\sigma^2}{N^3} \begin{bmatrix} 1/\alpha_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\alpha_n^2 \end{bmatrix}} \tag{4.3.9}$$

(See [Stoica and Nehorai 1989a; Stoica, Moses, Friedlander, and Söderström 1989].) In the case of Gaussian noise, the matrix in (4.3.9) can also be shown to equal the *Cramér–Rao limit matrix*, which gives a lower bound on the covariance matrix of any unbiased estimator of ω. (See Appendix B.) Hence, under the Gaussian hypothesis, the NLS method provides the most accurate (i.e., minimum variance) frequency estimates in a fairly general class of estimators. As a matter of fact, the variance of $\{\hat{\omega}_k\}$ (as given by (4.3.9)) often takes quite small values for reasonably large sample lengths N and signal-to-noise ratios $\text{SNR}_k = \alpha_k^2/\sigma^2$. For example, for $N = 300$ and $\text{SNR}_k = 30\text{dB}$, it follows from (4.3.9) that we may expect frequency estimation errors on the order of 10^{-5}, which is comparable with the roundoff errors in a 32-bit fixed-point processor.

The NLS method has another advantage that sets it apart from the subspace-based approaches that are discussed in the remainder of the chapter. The NLS method does not depend critically on the assumption that the noise process is white. If the noise process is not white, the NLS still gives consistent frequency estimates. In fact, the asymptotic covariance of the frequency estimates is diagonal, and $\text{var}(\hat{\omega}_k) = 6/(N^3\text{SNR}_k)$, where $\text{SNR}_k = \alpha_k^2/\phi_n(\omega_k)$ (here, $\phi_n(\omega)$ is the noise PSD) is the "local" signal-to-noise ratio of the sinusoid at frequency ω_k; see [Stoica and Nehorai 1989b], for example. Interestingly enough, the NLS method remains the most accurate method (if the data length is large) even in those cases where the (Gaussian) noise is colored [Stoica and Nehorai 1989b]. This fact spurred a renewed interest in the NLS approach and in reliable algorithms for performing the minimization required in (4.3.1); see, for example, [Hwang and Chen 1993; Ying, Potter, and Moses 1994; Li and Stoica 1996b; Umesh and Tufts 1996] and Complement 4.9.5.

Unfortunately, the good statistical performance associated with the NLS method of frequency estimation is difficult to achieve, for the following reason. The function (4.3.8) has a *complicated multimodal shape* with a *very sharp global maximum* corresponding to $\hat{\omega}$ [Stoica, Moses, Friedlander, and Söderström 1989]. Hence, finding $\hat{\omega}$ by a search algorithm requires very accurate initialization. Initialization procedures that provide fairly accurate approximations of the maximizer of (4.3.8) have been proposed in [Kumaresan, Scharf, and Shaw 1986], [Bresler and Macovski 1986], and [Ziskind and Wax 1988]. However, there is no available method which is guaranteed to provide frequency estimates within the attraction domain of the global maximum $\hat{\omega}$ of (4.3.8). As a consequence, a search algorithm could fail to converge to $\hat{\omega}$, or might even diverge.

The kinds of difficulties indicated above, which must be faced when using the NLS method in applications, limit the practical interest in this approach to frequency estimation. There are, however, some instances when the NLS approach may be turned into a practical frequency estimation method. Consider, first, the case of a single sinusoid ($n = 1$). A straightforward calculation shows that, in such a case, the first equation in (4.3.8) can be rewritten as

$$\hat{\omega} = \arg\max_{\omega} \hat{\phi}_p(\omega) \tag{4.3.10}$$

where $\hat{\phi}_p(\omega)$ is the periodogram (see (2.2.1))

$$\hat{\phi}_p(\omega) = \frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\omega t}\right|^2 \tag{4.3.11}$$

Hence, the NLS estimate of the frequency of a single sine wave buried in observation noise is given precisely by the highest peak of the unmodified periodogram.

Note that the above result is only approximately true (for $N \gg 1$) in the case of *real-valued* sinusoidal signals, a fact that lends additional support to the claim made in Chapter 1 that the analysis of the case of real-valued signals faces additional complications not encountered in the complex-valued case. Each real-valued sinusoid can be written as a sum of two complex exponentials, and the treatment of the real case with $n = 1$ is similar to that of the complex case with $n > 1$, presented next.

Next, consider the case of multiple sine waves ($n > 1$). The key condition that makes it possible to treat this case in a manner similar to the previous one is that the minimum frequency separation between the sine waves in the studied signal is larger than the periodogram's resolution limit:

$$\boxed{\Delta\omega = \inf_{k \neq p} |\omega_k - \omega_p| > 2\pi/N} \tag{4.3.12}$$

Since the estimation errors $\{\hat{\omega}_k - \omega_k\}$ from the NLS estimates are of order $\mathcal{O}(1/N^{3/2})$ (because $\text{cov}(\hat{\omega}) = \mathcal{O}(1/N^3)$; see (4.3.9)), equation (4.3.12) implies a similar inequality for the NLS frequency estimates $\{\hat{\omega}_k\}$: $\Delta\hat{\omega} > 2\pi/N$. It should then be possible to *resolve all n sine waves* in the noisy signal and to obtain *reasonable approximations* $\{\tilde{\omega}_k\}$ to $\{\hat{\omega}_k\}$ by evaluating the function in (4.3.8) at the points of a grid corresponding to the sampling of each frequency variable, as in the FFT:

$$\omega_k = \frac{2\pi}{N} j \qquad j = 0, \ldots, N-1 \qquad (k = 1, \ldots, n) \tag{4.3.13}$$

Of course, a direct application of such a grid method for the approximate maximization of (4.3.8) would be computationally burdensome for large values of n or N. However, it can be greatly simplified, as described next.

The p, k element of the matrix B^*B occurring in (4.3.8), when evaluated *at the points of the grid* (4.3.13), is given by

$$[B^*B]_{p,k} = N \qquad \text{for} \;\; p = k \tag{4.3.14}$$

and

$$\begin{aligned} [B^*B]_{p,k} &= \sum_{t=1}^{N} e^{i(\omega_k - \omega_p)t} = e^{i(\omega_k - \omega_p)} \frac{e^{iN(\omega_k - \omega_p)} - 1}{e^{i(\omega_k - \omega_p)} - 1} \\ &= 0 \qquad \text{for} \;\; p \neq k \end{aligned} \tag{4.3.15}$$

which implies that the function to be minimized in (4.3.8) has, in such a case, the following form:

$$\sum_{k=1}^{n} \frac{1}{N} \left| \sum_{t=1}^{N} y(t) e^{-i\omega_k t} \right|^2 \tag{4.3.16}$$

The previous additive decomposition in n functions of $\omega_1, \ldots, \omega_n$ (respectively) leads to the conclusion that $\{\tilde{\omega}_k\}$ (which, by definition, maximize (4.3.16) at the points of the grid (4.3.13)) are given by the n largest peaks of the periodogram. To show this, let us write the function in (4.3.16) as

$$g(\omega_1, \ldots, \omega_n) = \sum_{k=1}^{n} \hat{\phi}_p(\omega_k)$$

where $\hat{\phi}_p(\omega)$ is once again the periodogram. Observe that

$$\frac{\partial g(\omega_1, \ldots, \omega_n)}{\partial \omega_k} = \hat{\phi}_p'(\omega_k)$$

and

$$\frac{\partial^2 g(\omega_1, \ldots, \omega_n)}{\partial \omega_k \partial \omega_j} = \hat{\phi}_p''(\omega_k)\delta_{k,j}$$

Hence, the maximum points of (4.3.16) satisfy

$$\hat{\phi}_p'(\omega_k) = 0 \quad \text{and} \quad \hat{\phi}_p''(\omega_k) < 0 \quad \text{for } k = 1, \ldots, n$$

It follows that the set of maximizers of (4.3.16) is given by all possible combinations of n elements from the periodogram's peak locations. Now, recall the assumption made that $\{\omega_k\}$, and hence their estimates $\{\hat{\omega}_k\}$, are *distinct*. Under this assumption the highest maximum of $g(\omega_1, \ldots, \omega_n)$ is given by the locations of the n largest peaks of $\hat{\phi}_p(\omega)$, which is the desired result.

These findings are summarized as follows:

> Under the condition (4.3.12), the unmodified periodogram resolves all the n sine waves present in the noisy signal. Furthermore, the locations $\{\tilde{\omega}_k\}$ of the n largest peaks in the periodogram provide $\mathcal{O}(1/N)$ approximations to the NLS frequency estimates $\{\hat{\omega}_k\}$. In the case of $n = 1$, we have $\tilde{\omega}_1 = \hat{\omega}_1$ exactly. (4.3.17)

The fact that the differences $\{\tilde{\omega}_k - \hat{\omega}_k\}$ are $\mathcal{O}(1/N)$ means, of course, that the computationally convenient estimates $\{\tilde{\omega}_k\}$ (derived from the periodogram) will generally have an inflated variance compared to $\{\hat{\omega}_k\}$. However, $\{\tilde{\omega}_k\}$ can at least be used as initial values in a numerical implementation of the NLS estimator. In any case, this discussion indicates that, under (4.3.12), the periodogram performs quite well as a frequency estimator (which actually is the task for which it was introduced by Schuster more than a century ago!).

In the next sections, we present several "high-resolution" methods for frequency estimation, which exploit the *covariance matrix models*. More precisely, all of these methods derive frequency estimates by exploiting the properties of the eigendecomposition of data covariance matrices and, in particular, the subspaces associated with those matrices. For this reason, these methods are sometimes referred to by the generic name of *subspace methods*. However, in spite of their common subspace theme, the methods are quite different, and we will treat them in separate sections. The main features of these methods can be summarized as follows: (i) Their statistical performance is close to the ultimate performance corresponding to the NLS method (and given by the Cramér–Rao lower bound, (4.3.9)); (ii) unlike the NLS method, these methods are not based on multidimensional search procedures; and (iii) they do not depend on a "resolution condition," such as (4.3.12); thus, they could generally have a resolution threshold lower than that of the periodogram. The chief drawback of these methods, as compared with the NLS method, is that their performance significantly degrades if the measurement noise in (4.1.1) cannot be assumed to be white.

4.4 HIGH-ORDER YULE–WALKER METHOD

The high-order Yule–Walker (HOYW) method of frequency estimation can be derived from the *ARMA model* of the sinusoidal data, (4.2.3), much as can its counterpart in the rational PSD case. (See Section 3.7 and [Cadzow 1982; Stoica, Söderström, and Ti 1989; Stoica, Moses, Söderström, and Li 1991].) Actually, the HOYW method is based on an ARMA model of an order L *higher* than the minimal order n, for a reason that will be explained shortly.

If the polynomial $A(z)$ in (4.2.3) is multiplied by any other polynomial $\bar{A}(z)$, say of degree equal to $L - n$, then we obtain a higher order ARMA representation of our sinusoidal data, given by

$$y(t) + b_1 y(t-1) + \ldots + b_L y(t-L) = e(t) + b_1 e(t-1) + \ldots + b_L e(t-L) \tag{4.4.1}$$

or

$$B(z)y(t) = B(z)e(t)$$

where

$$B(z) = 1 + \sum_{k=1}^{L} b_k z^{-k} \triangleq A(z)\bar{A}(z) \tag{4.4.2}$$

Equation (4.4.1) can be rewritten in the following more condensed form (with obvious notation):

$$[y(t) \quad y(t-1) \ldots y(t-L)] \begin{bmatrix} 1 \\ b \end{bmatrix} = e(t) + \ldots + b_L e(t-L) \tag{4.4.3}$$

Premultiplying (4.4.3) by $[y^*(t-L-1) \ldots y^*(t-L-M)]^T$ and taking the expectation leads to

$$\Gamma^c \begin{bmatrix} 1 \\ b \end{bmatrix} = 0 \tag{4.4.4}$$

where the matrix Γ is defined in (4.2.8) and M is a positive integer that is yet to be specified. In order to obtain (4.4.4) as indicated previously, we made use of the fact that $E\{y^*(t-k)e(t)\} = 0$ for $k > 0$.

The similarity of (4.4.4) to the Yule–Walker system of equations encountered in Chapter 3 (see equation (3.7.1)) is more readily seen if (4.4.4) is rewritten in the following more detailed form:

$$\begin{bmatrix} r(L) & \dots & r(1) \\ \vdots & & \vdots \\ r(L+M-1) & \dots & r(M) \end{bmatrix} b = - \begin{bmatrix} r(L+1) \\ \vdots \\ r(L+M) \end{bmatrix} \tag{4.4.5}$$

Owing to this analogy, the set of equations (4.4.5) associated with the noisy sinusoidal signal $\{y(t)\}$ is said to form a HOYW system.

The HOYW matrix equation (4.4.4) can also be obtained directly from (4.2.8). For any $L \geq n$ and any polynomial $\bar{A}(z)$ (used in the defining equation, (4.4.2), for b), the elements of the vector

$$A_{L+1}^T \begin{bmatrix} 1 \\ b \end{bmatrix} \tag{4.4.6}$$

are equal to zero. Indeed, the kth row of (4.4.6) is

$$\begin{aligned} [1 \quad e^{-i\omega_k} \dots e^{-iL\omega_k}] \begin{bmatrix} 1 \\ b \end{bmatrix} &= 1 + \sum_{p=1}^{L} b_p e^{-i\omega_k p} \\ &= A(\omega_k)\bar{A}(\omega_k) = 0, \quad k = 1, \dots, n \end{aligned} \tag{4.4.7}$$

(since $A(\omega_k) = 0$, *cf.* (4.2.3)). It follows from (4.2.8) and (4.4.7) that the vector $\begin{bmatrix} 1 \\ b \end{bmatrix}$ lies in the *null space* of Γ^c (see Definition D2 in Appendix A),

$$\Gamma^c \begin{bmatrix} 1 \\ b \end{bmatrix} = 0$$

which is the desired result, (4.4.4).

The HOYW system of equations just derived can be used for frequency estimation in the following way: By replacing the unavailable theoretical covariances $\{r(k)\}$ in (4.4.5) by the sample covariances $\{\hat{r}(k)\}$, we obtain

$$\boxed{\begin{bmatrix} \hat{r}(L) & \dots & \hat{r}(1) \\ \vdots & & \vdots \\ \hat{r}(L+M-1) & \dots & \hat{r}(M) \end{bmatrix} \hat{b} \simeq - \begin{bmatrix} \hat{r}(L+1) \\ \vdots \\ \hat{r}(L+M) \end{bmatrix}} \tag{4.4.8}$$

Owing to the estimation errors in $\{\hat{r}(k)\}$, the matrix equation (4.4.8) cannot hold exactly in the general case, for any vector $\hat{b}$, as is indicated by the use of the "approximate equality" symbol $\simeq$.

We can solve (4.4.8) for $\hat{b}$ in a least-squares sense that is detailed in what follows, then form the polynomial

$$1 + \sum_{k=1}^{L} \hat{b}_k z^{-k} \tag{4.4.9}$$

and, finally (in view of (4.2.3) and (4.4.2)), obtain frequency estimates $\{\hat{\omega}_k\}$ as the angular positions of the n roots of (4.4.9) that are located nearest the unit circle.

It can be expected that increasing the values of M and L results in improved frequency estimates. Indeed, by increasing M and L we use higher lag covariances in (4.4.8), which could bear "additional information" on the data at hand. Increasing M and L also has a second, more subtle, effect that is explained next.

Let Ω denote the $M \times L$ covariance matrix in (4.4.5) and, similarly, let $\hat{\Omega}$ denote the sample covariance matrix in (4.4.8). It can be seen from (4.2.8) that

$$\operatorname{rank}(\Omega) = n \qquad \text{for } \; M, L \geq n \tag{4.4.10}$$

On the other hand, the matrix $\hat{\Omega}$ has full rank (almost surely)

$$\operatorname{rank}(\hat{\Omega}) = \min(M, L) \tag{4.4.11}$$

owing to the random errors in $\{\hat{r}(k)\}$. However, for reasonably large values of N, the matrix $\hat{\Omega}$ is close to the rank-n matrix Ω, since the sample covariances $\{\hat{r}(k)\}$ converge to $\{r(k)\}$ as N increases (as is shown in Complement 4.9.1). Hence, we may expect the linear system (4.4.8) to be *ill conditioned from a numerical standpoint*. (See the discussion in Section A.8.1 in Appendix A.) In fact, there is compelling empirical evidence that any LS procedure that estimates $\hat{b}$ directly from (4.4.8) has very poor accuracy. In order to overcome the previously described difficulty, we can make use of the *a priori rank information* (4.4.10). However, some preparations are required before we shall be able to do so. Let

$$\boxed{\hat{\Omega} = U \Sigma V^* \triangleq [\, \underbrace{U_1}_{n} \quad \underbrace{U_2}_{M-n} \,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \begin{matrix} \}\, n \\ \}\, L-n \end{matrix}} \tag{4.4.12}$$

denote the singular value decomposition (SVD) of the matrix $\hat{\Omega}$. (See Section A.4 in Appendix A; also [Söderström and Stoica 1989; Van Huffel and Vandewalle 1991] for general discussions on the SVD.) In (4.4.12), U is an $M \times M$ unitary matrix, V is an $L \times L$ unitary matrix, and Σ is an $M \times L$ diagonal matrix. $\hat{\Omega}$ is close to a rank-n matrix, so Σ_2 in (4.4.12) should be close to zero, which implies that

$$\hat{\Omega}_n \triangleq U_1 \Sigma_1 V_1^* \tag{4.4.13}$$

should be a good approximation for $\hat{\Omega}$. In fact, it can be proven that this $\hat{\Omega}_n$ is the *best* (in the Frobenius-norm sense) *rank-n approximation* of $\hat{\Omega}$ (Result R18 in Appendix A). Hence, in accordance with the rank information (4.4.10), we can use $\hat{\Omega}_n$ in (4.4.8) in lieu of $\hat{\Omega}$. The so-obtained *rank-truncated HOYW system of equations*

$$\hat{\Omega}_n \hat{b} \simeq - \begin{bmatrix} \hat{r}(L+1) \\ \vdots \\ \hat{r}(L+M) \end{bmatrix} \tag{4.4.14}$$

can be solved in a numerically sound way by using a simple LS procedure. It is readily verified that

$$\hat{\Omega}_n^\dagger = V_1 \Sigma_1^{-1} U_1^* \tag{4.4.15}$$

is the pseudoinverse of $\hat{\Omega}_n$. (See Definition D15 and Result R32.) Hence, the LS solution to (4.4.14) is given by

$$\boxed{\hat{b} = -V_1 \Sigma_1^{-1} U_1^* \begin{bmatrix} \hat{r}(L+1) \\ \vdots \\ \hat{r}(L+M) \end{bmatrix}} \tag{4.4.16}$$

The additional bonus for using $\hat{\Omega}_n$ instead of $\hat{\Omega}$ in (4.4.8) is an improvement in the statistical accuracy of the frequency estimates obtained from (4.4.16). This improved accuracy is explained by the fact that $\hat{\Omega}_n$ should be closer to Ω than $\hat{\Omega}$ is; the improved covariance matrix estimate $\hat{\Omega}_n$ obtained by exploitation of the rank information (4.4.10), when used in the HOYW system of equations, should lead to refined frequency estimates.

We remark that a *total least-squares* (TLS) solution for $\hat{b}$ can also be obtained from (4.4.8). (See Definition D17 and Result R33 in Appendix A.) A TLS solution makes sense, because we have errors in both $\hat{\Omega}$ and the right-hand-side vector in equation (4.4.8). In fact the TLS-based estimate of b is often slightly better than the estimate discussed above, which is obtained as the LS solution to the *rank-truncated* system of linear equations in (4.4.14).

We next return to the selection of L and M. As M and L increase, the information brought into the estimation problem under study by the rank condition (4.4.10) is more and more important, and hence the corresponding increase of accuracy is more and more pronounced. (For instance, the information that a 10×10 noisy matrix has rank one in the noise-free case leads to more relations between the matrix elements, and hence to more "noise cleaning," than if the matrix were 2×2.) In fact, for $M = n$ or $L = n$, the rank condition is inactive; $\hat{\Omega}_n = \hat{\Omega}$ in such a case. The previous discussion gives another explanation of why the accuracy of the frequency estimates obtained from (4.4.16) may be expected to increase with increasing M and L.

The next box summarizes the *HOYW frequency estimation method.* It should be noted that the operation in Step 3 of the HOYW method is implicitly based on the assumption that the estimated "*signal roots*" (i.e., the roots of $A(z)$ in (4.4.2)) are always closer to the unit circle than the estimated "*noise roots*" (i.e., the roots of $\bar{A}(z)$ in (4.4.2)). It can be shown that as $N \to \infty$, all roots of $\bar{A}(z)$ are strictly inside the unit circle; see, for example, Complement 6.5.1 and [Kumaresan and Tufts 1983]. This property cannot be guaranteed in finite samples, but there is empirical evidence that it holds quite often. In those rare cases where it fails to hold, the HOYW method produces *spurious* (or *false*) *frequency estimates*. The risk of producing spurious estimates is the price paid for the improved accuracy obtained by increasing L. (Note that, for $L = n$, there is no "noise root," and hence no spurious estimate can occur in such a case.) The risk of false frequency estimation is a problem that is common to all methods that estimate the frequencies from the roots of a polynomial of degree larger than n, such as the MUSIC and Min–Norm methods, to be discussed in the next two sections.

The HOYW Frequency Estimation Method

Step 1. Compute the sample covariances $\{\hat{r}(k)\}_{k=1}^{L+M}$. We may set $L \simeq M$ and select the values of these integers so that $L + M$ is a fraction of the sample length (such as $N/3$). Note that, if $L + M$ is set to a value too close to N, then the higher lag covariances required in (4.4.8) cannot be estimated in a reliable way.

Step 2. Compute the SVD of $\hat{\Omega}$, (4.4.12), and compute $\hat{b}$ by using (4.4.16).

Step 3. Isolate the n roots of the polynomial (4.4.9) that are closest to the unit circle and obtain the frequency estimates as the angular positions of these roots.

4.5 PISARENKO AND MUSIC METHODS

The *MUltiple SIgnal Classification* (or *MUltiple SIgnal Characterization*) (MUSIC) method [Schmidt 1979; Bienvenu 1979] and Pisarenko's method [Pisarenko 1973] (a special case of MUSIC, as is explained next) are derived from the covariance model (4.2.7) with $m > n$. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ denote the *eigenvalues* of R in (4.2.7), arranged in nonincreasing order, and let $\{s_1, \ldots, s_n\}$ be the *orthonormal eigenvectors* associated with $\{\lambda_1, \ldots, \lambda_n\}$, and $\{g_1, \ldots, g_{m-n}\}$ a set of *orthonormal eigenvectors* corresponding to $\{\lambda_{n+1}, \ldots, \lambda_m\}$. (See Appendix A.) Since

$$\operatorname{rank}(APA^*) = n \tag{4.5.1}$$

it follows that APA^* has n strictly positive eigenvalues, the remaining $(m - n)$ eigenvalues all being equal to zero. Combining this observation with the fact that (see Result R5 in Appendix A)

$$\lambda_k = \tilde{\lambda}_k + \sigma^2 \qquad (k = 1, \ldots, m) \tag{4.5.2}$$

where $\{\tilde{\lambda}_k\}_{k=1}^m$ are the eigenvalues of APA^* arranged in nonincreasing order, leads to the following result:

$$\begin{cases} \lambda_k > \sigma^2 & \text{for } k = 1, \ldots, n \\ \lambda_k = \sigma^2 & \text{for } k = n+1, \ldots, m \end{cases} \tag{4.5.3}$$

The set of eigenvalues of R can hence be split into two subsets. Next, we show that the eigenvectors associated with each of these subsets, as introduced previously, possess some interesting properties that can be used for frequency estimation.

Let

$$S = [s_1, \ldots, s_n] \quad (m \times n), \qquad G = [g_1, \ldots, g_{m-n}] \quad (m \times (m-n)) \tag{4.5.4}$$

From (4.2.7) and (4.5.3), we get at once

$$RG = G \begin{bmatrix} \lambda_{n+1} & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} = \sigma^2 G = APA^*G + \sigma^2 G \tag{4.5.5}$$

The first equality in (4.5.5) follows from the definition of G and $\{\lambda_k\}_{k=n+1}^{m}$, the second equality follows from (4.5.3), and the third from (4.2.7). The last equality in equation (4.5.5) implies that $APA^*G = 0$, or (as the matrix AP has full column rank)

$$\boxed{A^*G = 0} \tag{4.5.6}$$

In other words, the columns $\{g_k\}$ of G belong to the *null space* of A^*, a fact which is denoted by $g_k \in \mathcal{N}(A^*)$. Since $\operatorname{rank}(A) = n$, the dimension of $\mathcal{N}(A^*)$ is equal to $m - n$, which is also the dimension of the *range space* of G, $\mathcal{R}(G)$. It follows from this observation and (4.5.6) that

$$\boxed{\mathcal{R}(G) = \mathcal{N}(A^*)} \tag{4.5.7}$$

In words, (4.5.7) says that the vectors $\{g_k\}$ span both $\mathcal{R}(G)$ and $\mathcal{N}(A^*)$. Now, by definition,

$$S^*G = 0 \tag{4.5.8}$$

so we also have $\mathcal{R}(G) = \mathcal{N}(S^*)$; hence, $\mathcal{N}(S^*) = \mathcal{N}(A^*)$. Since $\mathcal{R}(S)$ and $\mathcal{R}(A)$ are the orthogonal complements to $\mathcal{N}(S^*)$ and $\mathcal{N}(A^*)$, it follows that

$$\boxed{\mathcal{R}(S) = \mathcal{R}(A)} \tag{4.5.9}$$

We can also derive the equality (4.5.9) directly from (4.2.7). Set

$$\mathring{\Lambda} = \begin{bmatrix} \lambda_1 - \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \lambda_n - \sigma^2 \end{bmatrix} \tag{4.5.10}$$

From

$$RS = S \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} = APA^*S + \sigma^2 S \tag{4.5.11}$$

we obtain

$$S = A\left(PA^*S\,\mathring{\Lambda}^{-1}\right) \tag{4.5.12}$$

which shows that $\mathcal{R}(S) \subset \mathcal{R}(A)$. However, $\mathcal{R}(S)$ and $\mathcal{R}(A)$ have the same dimension (equal to n); hence, (4.5.9) follows. Owing to (4.5.9) and (4.5.8), the subspaces $\mathcal{R}(S)$ and $\mathcal{R}(G)$ are sometimes called the *signal subspace* and *noise subspace*, respectively.

The following key result is obtained from (4.5.6):

$$\boxed{\begin{array}{c}\text{The true frequency values } \{\omega_k\}_{k=1}^n \text{ are the only solutions of the equation} \\ a^*(\omega)GG^*a(\omega) = 0 \text{ for any } m > n.\end{array}} \tag{4.5.13}$$

The fact that $\{\omega_k\}$ satisfy this equation follows from (4.5.6). It only remains to prove that $\{\omega_k\}_{k=1}^n$ are the only solutions to (4.5.13). Let $\tilde{\omega}$ denote another possible solution, with $\tilde{\omega} \neq \omega_k$ $(k = 1, \ldots, n)$. In (4.5.13), GG^* is the *orthogonal projector* onto $\mathcal{R}(G)$. (See Section A.4.) Hence, (4.5.13) implies that $a(\tilde{\omega})$ is orthogonal to $\mathcal{R}(G)$, which means that $a(\tilde{\omega}) \in \mathcal{N}(G^*)$. However, the Vandermonde vector $a(\tilde{\omega})$ is linearly independent of $\{a(\omega_k)\}_{k=1}^n$. Since $n + 1$ linearly independent vectors cannot belong to an n-dimensional subspace, which is $\mathcal{N}(G^*)$ in the present case, we conclude that no other solution $\tilde{\omega}$ to (4.5.13) can exist; with this, the proof is finished.

The *MUSIC algorithm* uses the previous result to derive frequency estimates in the following steps:

Step 1. Compute the sample covariance matrix

$$\hat{R} = \frac{1}{N}\sum_{t=m}^{N} \tilde{y}(t)\tilde{y}^*(t) \tag{4.5.14}$$

and its eigendecomposition. Let $\hat{S}$ and $\hat{G}$ denote the matrices defined similarly to S and G, but made from the eigenvectors $\{\hat{s}_1, \ldots, \hat{s}_n\}$ and $\{\hat{g}_1, \ldots, \hat{g}_{m-n}\}$ of $\hat{R}$.

Step 2a. *(Spectral MUSIC)* [Schmidt 1979; Bienvenu 1979]. Determine frequency estimates as the locations of the n highest peaks of the function

$$\frac{1}{a^*(\omega)\hat{G}\hat{G}^*a(\omega)}, \qquad \omega \in [-\pi, \pi] \tag{4.5.15}$$

(Sometimes (4.5.15) is called a "*pseudospectrum*," since it indicates the presence of sinusoidal components in the studied signal, but it is not a true PSD. This fact may explain the attribute "spectral" attached to this variant of MUSIC.)

OR

Step 2b. *(Root MUSIC)* [Barabell 1983]. Determine frequency estimates as the angular positions of the n (pairs of reciprocal) roots of the equation

$$a^T(z^{-1})\hat{G}\hat{G}^*a(z) = 0 \tag{4.5.16}$$

which are located nearest the unit circle. In (4.5.16), $a(z)$ stands for the vector $a(\omega)$, (4.2.4), with $e^{i\omega}$ replaced by z, so

$$a(z) = [1, z^{-1}, \ldots, z^{-(m-1)}]^T$$

For $m = n + 1$ (which is the *minimum* possible value), the MUSIC algorithm reduces to the Pisarenko method, which was the earliest proposal for an eigenanalysis-based (or subspace-based) method of frequency estimation [Pisarenko 1973]:

$$\boxed{\text{The Pisarenko method is MUSIC with } m = n + 1.} \tag{4.5.17}$$

In the Pisarenko method, the estimated frequencies are computed from (4.5.16). For $m = n + 1$, this $2(m - 1)$-degree equation can be reduced to the following equation of degree $m - 1 = n$:

$$a^T(z^{-1})\hat{g}_1 = 0 \tag{4.5.18}$$

The Pisarenko frequency estimates are obtained as the angular positions of the roots of (4.5.18). The Pisarenko method is the simplest version of MUSIC from a computational standpoint. In addition, unlike MUSIC with $m > n + 1$, the Pisarenko procedure does not have the problem of separating the "signal roots" from the "noise roots." (See the discussion on this point at the end of Section 4.4.) However, it can be shown that *the accuracy of the MUSIC frequency estimates increases significantly with increasing m*. Hence, the price paid for the computational simplicity of the Pisarenko method could be a relatively poor statistical accuracy.

Regarding the *selection of a value for m*, this parameter may be chosen as large as possible, but not too close to N, in order to still allow a reliable estimation of the covariance matrix (for example, as in (4.5.14)). In some applications, the largest possible value that may be selected for m may also be limited by computational complexity considerations.

Whenever the *tradeoff between statistical accuracy and computational complexity* is an important issue, the following simple ideas can be valuable.

The *finite-sample statistical accuracy* of MUSIC frequency estimates may be improved by modifying the covariance estimator (4.5.14). For instance, $\hat{R}$ is not Toeplitz, whereas the true covariance matrix R is. We may correct this situation by replacing the elements in each diagonal of $\hat{R}$ with their average. The so-corrected sample covariance matrix can be shown to be the best (in the Frobenius-norm sense) Toeplitz approximation of $\hat{R}$. Another modification of $\hat{R}$, with the same purpose of improving the finite-sample statistical accuracy, is described in Section 4.8.

The *computational complexity* of MUSIC, for a given m, can be reduced in various ways. Quite often, m is such that $m - n > n$. Then, the computational burdens associated with both Spectral and Root MUSIC can be reduced by using $I - \hat{S}\hat{S}^*$ in (4.5.15) or (4.5.16) in lieu of $\hat{G}\hat{G}^*$. (Note that $\hat{S}\hat{S}^* + \hat{G}\hat{G}^* = I$ by the very definition of the eigenvector matrices.) The computational burden of Root MUSIC can be further reduced as explained next. The polynomial in (4.5.16) is a self-reciprocal (or symmetric) one: its roots appear in reciprocal pairs $(\rho e^{i\varphi}, \frac{1}{\rho}e^{i\varphi})$. On the unit circle $z = e^{i\omega}$, (4.5.16) is nonnegative and, hence, may be interpreted as a PSD. These properties mentioned imply that (4.5.16) can be factored as

$$a^T(z^{-1})\hat{G}\hat{G}^*a(z) = \alpha(z)\alpha^*(1/z^*) \tag{4.5.19}$$

where $\alpha(z)$ is a polynomial of degree $(m - 1)$ with all its zeroes located within or on the unit circle. We can then find the frequency estimates from the n roots of $\alpha(z)$ that are closest to the unit circle. Since there are efficient numerical procedures for spectral factorization, determining $\alpha(z)$, as in (4.5.19), and then computing its zeroes is usually computationally more efficient than finding the (reciprocal) roots of the $2(m - 1)$-degree polynomial (4.5.16).

Finally, we address the issue of *spurious frequency estimates*. As implied by the result (4.5.13), for $N \to \infty$ there is no risk of obtaining false frequency estimates. However, in finite samples, such a risk always exists. Usually, this risk is quite small but it could become a real problem if m takes on large values. The key result on which the standard MUSIC algorithm, (4.5.15), is based can be used to derive a *modified MUSIC* that does not suffer from the spurious-estimate problem. In what follows, we explain only the basic ideas leading to the modified MUSIC method, without going into details of its implementation. (For such details, the interested reader may consult [Stoica and Sharman 1990].) Let $\{c_k\}_{k=1}^n$ denote the coefficients of the polynomial $A(z)$ defined in (4.2.3); that is,

$$A(z) = 1 + c_1z^{-1} + \ldots + c_nz^{-n} = \prod_{k=1}^{n}(1 - e^{i\omega_k}z^{-1}) \tag{4.5.20}$$

Introduce the following matrix made from $\{c_k\}$:

$$C^* = \begin{bmatrix} 1 & c_1 & \ldots & c_n & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & c_1 & \ldots & c_n \end{bmatrix}, \qquad (m-n) \times m \tag{4.5.21}$$

It is readily verified that

$$C^*A = 0, \qquad (m-n) \times n \tag{4.5.22}$$

where A is defined in (4.2.4). Combining (4.5.9) and (4.5.22) gives

$$C^*S = 0, \qquad (m-n) \times n \tag{4.5.23}$$

which is the key property here. The matrix equation (4.5.23) can be rewritten in the form

$$\phi c = \mu \tag{4.5.24}$$

where the $(m-n)n \times n$ matrix ϕ and the $(m-n)n \times 1$ vector μ are entirely determined from the elements of S, and where

$$c = [c_1 \dots c_n]^T \tag{4.5.25}$$

By replacing the elements of S in ϕ and μ by the corresponding entries of $\hat{S}$, we obtain the sample version of (4.5.24),

$$\hat{\phi}\hat{c} \simeq \hat{\mu} \tag{4.5.26}$$

from which an estimate $\hat{c}$ of c may be obtained by an LS or TLS algorithm; see Section A.8 for details. The frequency estimates can then be derived from the roots of the estimated polynomial (4.5.20) corresponding to $\hat{c}$. Since this polynomial has a (minimal) degree equal to n, there is *no risk* for false frequency estimation.

4.6 MIN–NORM METHOD

MUSIC uses $(m-n)$ linearly independent vectors in $\mathcal{R}(\hat{G})$ to obtain the frequency estimates. Since any vector in $\mathcal{R}(\hat{G})$ is (asymptotically) orthogonal to $\{a(\omega_k)\}_{k=1}^{n}$ (*cf.* (4.5.7)), we may think of using *only one* such vector for frequency estimation. By doing so, we might achieve some computational saving, hopefully without sacrificing too much accuracy.

The Min–Norm method proceeds to estimate the frequencies along these lines [KUMARESAN AND TUFTS 1983]. Let

$$\boxed{\begin{bmatrix} 1 \\ \hat{g} \end{bmatrix} = \begin{array}{l}\text{the vector in } \mathcal{R}(\hat{G}), \text{ with first element equal to one,} \\ \text{that has minimum Euclidean norm.}\end{array}} \tag{4.6.1}$$

Then, the *Min–Norm frequency estimates* are determined as

(*Spectral Min–Norm*). The locations of the n highest peaks in the pseudospectrum

$$\frac{1}{\left| a^*(\omega) \begin{bmatrix} 1 \\ \hat{g} \end{bmatrix} \right|^2} \tag{4.6.2}$$

or, alternatively,

(*Root Min–Norm*). The angular positions of the n roots of the polynomial

$$a^T(z^{-1})\begin{bmatrix} 1 \\ \hat{g} \end{bmatrix} \tag{4.6.3}$$

that are located nearest to the unit circle.

It remains to find the vector in (4.6.1) and, in particular, to show that its first element can always be normalized to 1. We will later comment on the reason behind the specific selection (4.6.1) of a vector in $\mathcal{R}(\hat{G})$. In the following, the Euclidean norm of a vector is denoted by $\|\cdot\|$.

Partition the matrix $\hat{S}$ as

$$\hat{S} = \begin{bmatrix} \alpha^* \\ \bar{S} \end{bmatrix} \begin{matrix} \}\,1 \\ \}\,m-1 \end{matrix} \tag{4.6.4}$$

As $\begin{bmatrix} 1 \\ \hat{g} \end{bmatrix} \in \mathcal{R}(\hat{G})$, it must satisfy the equation

$$\hat{S}^*\begin{bmatrix} 1 \\ \hat{g} \end{bmatrix} = 0 \tag{4.6.5}$$

which, by using (4.6.4), can be rewritten as

$$\bar{S}^*\hat{g} = -\alpha \tag{4.6.6}$$

The minimum–norm solution to (4.6.6) is given (see Result R31 in Appendix A) by

$$\hat{g} = -\bar{S}(\bar{S}^*\bar{S})^{-1}\alpha \tag{4.6.7}$$

assuming that the inverse exists. Note that

$$I = \hat{S}^*\hat{S} = \alpha\alpha^* + \bar{S}^*\bar{S} \tag{4.6.8}$$

and also that one eigenvalue of $I - \alpha\alpha^*$ is equal to $1 - \|\alpha\|^2$ and the remaining $(n-1)$ eigenvalues of $I - \alpha\alpha^*$ are equal to 1; it follows that the inverse in (4.6.7) exists if and only if

$$\|\alpha\|^2 \neq 1 \tag{4.6.9}$$

If this condition is not satisfied, there will be no vector of the form of (4.6.1) in $\mathcal{R}(\hat{G})$. We postpone the study of (4.6.9) until we obtain a final-form expression for $\hat{g}$.

Under the condition (4.6.9), a simple calculation shows that

$$(\bar{S}^*\bar{S})^{-1}\alpha = (I - \alpha\alpha^*)^{-1}\alpha = \alpha/(1 - \|\alpha\|^2) \tag{4.6.10}$$

Inserting (4.6.10) in (4.6.7) gives

$$\boxed{\hat{g} = -\bar{S}\alpha/(1 - \|\alpha\|^2)} \tag{4.6.11}$$

which expresses $\hat{g}$ as a function of the elements of $\hat{S}$.

We can also obtain $\hat{g}$ as a function of the entries in $\hat{G}$. To do so, partition $\hat{G}$ as

$$\hat{G} = \begin{bmatrix} \beta^* \\ \bar{G} \end{bmatrix} \tag{4.6.12}$$

Since $\hat{S}\hat{S}^* = I - \hat{G}\hat{G}^*$ by the definition of the matrices $\hat{S}$ and $\hat{G}$, it follows that

$$\begin{bmatrix} \|\alpha\|^2 & (\bar{S}\alpha)^* \\ \bar{S}\alpha & \bar{S}\bar{S}^* \end{bmatrix} = \begin{bmatrix} 1 - \|\beta\|^2 & -(\bar{G}\beta)^* \\ -\bar{G}\beta & I - \bar{G}\bar{G}^* \end{bmatrix} \tag{4.6.13}$$

Comparing the blocks in (4.6.13) makes it possible to express $\|\alpha\|^2$ and $\bar{S}\alpha$ as functions of $\bar{G}$ and β, which leads to the following equivalent expression for $\hat{g}$:

$$\boxed{\hat{g} = \bar{G}\beta/\|\beta\|^2} \tag{4.6.14}$$

If $m - n > n$, then it is computationally more advantageous to obtain $\hat{g}$ from (4.6.11); *otherwise*, (4.6.14) should be used.

Next, we return to the condition (4.6.9), which is implicitly assumed to hold in the previous derivations. As already mentioned, this condition is equivalent to rank$(\bar{S}^*\bar{S}) = n$ which, in turn, holds if and only if

$$\text{rank}(\bar{S}) = n \tag{4.6.15}$$

Now, it follows from (4.5.9) that any block of S made from more than n consecutive rows should have rank equal to n. Hence, (4.6.15) must hold at least for N sufficiently large. With this observation, the derivation of the Min–Norm frequency estimator is complete.

The statistical accuracy of the Min–Norm method is similar to that corresponding to MUSIC. Hence, Min–Norm achieves MUSIC's performance at a reduced computational cost. It should be noted that the selection (4.6.1) of the vector in $\mathcal{R}(\hat{G})$, used in the Min–Norm algorithm, is critical in obtaining frequency estimates with satisfactory statistical accuracy. Other choices of vectors in $\mathcal{R}(\hat{G})$ could give rather poor accuracy. In addition, there is empirical evidence that the use of the minimum–norm vector in $\mathcal{R}(\hat{G})$, as in (4.6.1), can decrease the risk of spurious frequency estimates, as compared with the use of other vectors in $\mathcal{R}(\hat{G})$ or even with MUSIC. See Complement 6.5.1 for details on this aspect.

4.7 ESPRIT METHOD

Let

$$A_1 = [I_{m-1} \quad 0]A \qquad (m-1) \times n \tag{4.7.1}$$

and

$$A_2 = [0 \quad I_{m-1}]A \qquad (m-1) \times n \tag{4.7.2}$$

where I_{m-1} is the identity matrix of dimension $(m-1) \times (m-1)$ and $[I_{m-1} \quad 0]$ and $[0 \quad I_{m-1}]$ are $(m-1) \times m$. It is readily verified that

$$A_2 = A_1 D \tag{4.7.3}$$

where

$$D = \begin{bmatrix} e^{-i\omega_1} & & 0 \\ & \ddots & \\ 0 & & e^{-i\omega_n} \end{bmatrix} \tag{4.7.4}$$

D is a unitary matrix, so the transformation in (4.7.3) is a *rotation*. ESPRIT (*Estimation of Signal Parameters by Rotational Invariance Techniques*: [Paulraj, Roy, and Kailath 1986; Roy and Kailath 1989]; see also [Kung, Arun, and Rao 1983]), relies on the rotational transformation (4.7.3), as we detail next.

Similarly to (4.7.1) and (4.7.2), define

$$S_1 = [I_{m-1} \quad 0]S \tag{4.7.5}$$

$$S_2 = [0 \quad I_{m-1}]S \tag{4.7.6}$$

From (4.5.12), we have that

$$S = AC \tag{4.7.7}$$

where C is the $n \times n$ nonsingular matrix given by

$$C = PA^* S \mathring{\Lambda}^{-1} \tag{4.7.8}$$

(Observe that both S and A in (4.7.7) have full column rank, and hence, C must be nonsingular; see Result R2 in Appendix A.) The foregoing explicit expression for C actually has no relevance to the present discussion. It is only (4.7.7) and the fact that C is nonsingular that count.

By using (4.7.1)–(4.7.3) and (4.7.7), we can write

$$S_2 = A_2 C = A_1 DC = S_1 C^{-1} DC = S_1 \phi \tag{4.7.9}$$

where

$$\phi \triangleq C^{-1}DC \tag{4.7.10}$$

The Vandermonde structure of A, implies that the matrices A_1 and A_2 have full column rank (equal to n). In view of (4.7.7), S_1 and S_2 must also have full column rank. It then follows from (4.7.9) that the matrix ϕ is given uniquely by

$$\phi = (S_1^* S_1)^{-1} S_1^* S_2 \tag{4.7.11}$$

This formula expresses ϕ as a function of some quantities that can be estimated from the available sample. The importance of being able to estimate ϕ stems from the fact that ϕ and D have the same eigenvalues. (This can be seen from equation (4.7.10), which is a *similarity transformation* relating ϕ and D, along with Result R6 in Appendix A.)

ESPRIT uses the previous observations to compute frequency estimates as described here:

> ESPRIT estimates the frequencies $\{\omega_k\}_{k=1}^{n}$ as $-\arg(\hat{\nu}_k)$, where $\{\hat{\nu}_k\}_{k=1}^{n}$ are the eigenvalues of the following (consistent) estimate of the matrix ϕ:
>
> $$\hat{\phi} = (\hat{S}_1^* \hat{S}_1)^{-1} \hat{S}_1^* \hat{S}_2$$

(4.7.12)

It should be noted that this estimate of ϕ is implicitly obtained by solving the linear system of equations

$$\hat{S}_1 \hat{\phi} \simeq \hat{S}_2 \tag{4.7.13}$$

by an *LS method*. It has been empirically observed that better finite-sample accuracy might be achieved if (4.7.13) is solved for $\hat{\phi}$ by a *Total LS method*. (See Section A.8 and [Van Huffel and Vandewalle 1991] for discussions on the TLS approach.)

The *statistical accuracy* of ESPRIT is similar to that of the previously described methods: HOYW, MUSIC, and Min–Norm. In fact, in most cases, ESPRIT may provide slightly more accurate frequency estimates than do the other methods mentioned, yet at similar computational cost. In addition, unlike these other methods, ESPRIT has *no problem* with separating the "signal roots" from the "noise roots," as can be seen from (4.7.12). Note that this property is shared by the modified MUSIC method (discussed in Section 4.5); however, in many cases, ESPRIT outperforms modified MUSIC in terms of statistical accuracy. All these considerations recommend ESPRIT as the first choice in a frequency estimation application.

4.8 FORWARD–BACKWARD APPROACH

The previously described eigenanalysis-based methods (MUSIC, Min–Norm, and ESPRIT) derive their frequency estimates from the eigenvectors of the sample covariance matrix $\hat{R}$, (4.5.14), which is restated here for easy reference:

$$\hat{R} = \frac{1}{N}\sum_{t=m}^{N} \begin{bmatrix} y(t) \\ \vdots \\ y(t-m+1) \end{bmatrix} [y^*(t) \dots y^*(t-m+1)] \tag{4.8.1}$$

$\hat{R}$ is recognized to be the matrix that appears in the least-squares (LS) estimation of the coefficients $\{\alpha_k\}$ of an mth-order *forward* linear predictor of $y^*(t+1)$:

$$\hat{y}^*(t+1) = \alpha_1 y^*(t) + \dots + \alpha_m y^*(t-m+1) \tag{4.8.2}$$

For this reason, the methods that obtain frequency estimates from $\hat{R}$ are named *forward (F) approaches*.

Extensive numerical experience with the aforementioned methods has shown that the corresponding frequency-estimation accuracy can be enhanced by using, in lieu of $\hat{R}$, the modified sample covariance matrix

$$\boxed{\tilde{R} = \frac{1}{2}(\hat{R} + J\hat{R}^T J)} \tag{4.8.3}$$

where

$$J = \begin{bmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{bmatrix} \tag{4.8.4}$$

is the so-called "*exchange*" (or "*reversal*") *matrix*. The second term in (4.8.3) has the following detailed form:

$$J\hat{R}^T J = \frac{1}{N}\sum_{t=m}^{N} \begin{bmatrix} y^*(t-m+1) \\ \vdots \\ y^*(t) \end{bmatrix} [y(t-m+1) \dots y(t)] \tag{4.8.5}$$

The matrix (4.8.5) is the one that appears in the LS estimate of the coefficients of an mth-order *backward* linear predictor of $y(t-m)$:

$$\hat{y}(t-m) = \mu_1 y(t-m+1) + \dots + \mu_m y(t) \tag{4.8.6}$$

This observation, along with the previous remark made about $\hat{R}$, suggests the name *forward–backward (FB) approaches* for methods that obtain frequency estimates from $\tilde{R}$ in (4.8.3).

The (i, j) element of $\tilde{R}$ is given by

$$\begin{aligned}\tilde{R}_{i,j} &= \frac{1}{2N}\sum_{t=m}^{N}[y(t-i)y^*(t-j) + y^*(t-m+1+i)y(t-m+1+j)] \\ &\triangleq T_1 + T_2 \qquad (i, j = 0, \ldots, m-1)\end{aligned} \tag{4.8.7}$$

Assume that $i \leq j$ (the other case $i \geq j$ can be similarly treated). Let $\hat{r}(j-i)$ denote the usual sample covariance:

$$\hat{r}(j-i) = \frac{1}{N}\sum_{t=(j-i)+1}^{N} y(t)y^*(t-(j-i)) \tag{4.8.8}$$

A straightforward calculation shows that the two terms T_1 and T_2 in (4.8.7) can be written as

$$T_1 = \frac{1}{2N}\sum_{p=m-i}^{N-i} y(p)y^*(p-(j-i)) = \frac{1}{2}\hat{r}(j-i) + \mathcal{O}(1/N) \tag{4.8.9}$$

and

$$T_2 = \frac{1}{2N}\sum_{p=j+1}^{N-m+j+1} y(p)y^*(p-(j-i)) = \frac{1}{2}\hat{r}(j-i) + \mathcal{O}(1/N) \tag{4.8.10}$$

where $\mathcal{O}(1/N)$ denotes a term that tends to zero as $1/N$ when N increases (it is here assumed that $m \ll N$). It follows from (4.8.7)–(4.8.10) that, for large N, the difference between $\tilde{R}_{i,j}$ or $\hat{R}_{i,j}$ and the sample covariance lag $\hat{r}(j-i)$ is "small." Hence, the frequency estimation methods based on $\hat{R}$ or $\tilde{R}$ (or on $[\hat{r}(j-i)]$) may be expected to have similar performances in large samples.

In summary, it follows from the previous discussion that the empirically observed performance superiority of the forward–backward approach over the forward-only approach should only be manifest in samples with relatively small lengths. As such, this superiority cannot easily be established by theoretical means. Let us then argue heuristically.

First, note that the transformation $J(\cdot)^T J$ is such that the following equalities hold:

$$(\hat{R})_{i,j} = (J\hat{R}J)_{m-i,m-j} = (J\hat{R}^T J)_{m-j,m-i} \tag{4.8.11}$$

and

$$(\hat{R})_{m-j,m-i} = (J\hat{R}^T J)_{i,j} \tag{4.8.12}$$

This implies that the (i, j) and $(m-j, m-i)$ elements of $\tilde{R}$ are both given by

$$\tilde{R}_{i,j} = \tilde{R}_{m-j,m-i} = \frac{1}{2}(\hat{R}_{i,j} + \hat{R}_{m-j,m-i}) \tag{4.8.13}$$

Equations (4.8.11)–(4.8.12) imply that $\tilde{R}$ is invariant to the transformation $J(\cdot)^T J$:

$$J\tilde{R}^T J = \tilde{R} \tag{4.8.14}$$

Such a matrix is said to be *persymmetric* (or *centrosymmetric*). In order to see the reason for this name, note that $\tilde{R}$ is Hermitian (symmetric in the real-valued case) with respect to its main diagonal; *in addition*, $\tilde{R}$ is symmetric about its main antidiagonal. Indeed, the equal elements $\tilde{R}_{i,j}$ and $\tilde{R}_{m-j,m-i}$ of $\tilde{R}$ belong to the same diagonal as $i - j = (m - j) - (m - i)$. They are also symmetrically placed with respect to the main antidiagonal; $\tilde{R}_{i,j}$ lies on antidiagonal $(i + j)$, $\tilde{R}_{m-j,m-i}$ on the $[2m - (j + i)]$th one, and the main antidiagonal is the mth one (and $m = [(i + j) + 2m - (i + j)]/2$).

The theoretical (and unknown) covariance matrix R is Toeplitz and hence persymmetric. Since $\tilde{R}$ is persymmetric like R, whereas $\hat{R}$ is not, we may expect $\tilde{R}$ to be a better estimate of R than $\hat{R}$. In turn, this means that the frequency estimates derived from $\tilde{R}$ are likely to be more accurate than those obtained from $\hat{R}$.

The impact of enforcing the persymmetric property can be seen by examining, say, the $(1, 1)$ and (m, m) elements of $\hat{R}$ and $\tilde{R}$. Both the $(1, 1)$ and (m, m) elements of $\hat{R}$ are estimates of $r(0)$; however, the $(1, 1)$ element does not use the first $(m - 1)$ lag products $|y(1)|^2, \ldots, |y(m - 1)|^2$, and the (m, m) element does not use the last $(m - 1)$ lag products $|y(N - m + 2)|^2, \ldots, |y(N)|^2$. If $N \gg m$, the omission of these lag products is negligible; for small N, however, this omission can be significant. On the other hand, all lag products of $y(t)$ are used to form the $(1, 1)$ and (m, m) elements of $\tilde{R}$, and, in general, the (i, j) element of $\tilde{R}$ uses more lag products of $y(t)$ than does the corresponding element of $\hat{R}$. (For more details on the FB approach, we refer the reader to, e.g., [Rao and Hari 1993; Pillai 1989]; see also Complement 6.5.8.)

Finally, the reader might wonder why we do not replace $\hat{R}$ by a Toeplitz estimate, obtained (for example) by averaging the elements along each diagonal of $\hat{R}$. This Toeplitz estimate would at first seem to be a better approximation of R than either $\hat{R}$ or $\tilde{R}$. The reason why we do not “Toeplitz-ize” $\hat{R}$ or $\tilde{R}$ is that, for finite N and infinite signal-to-noise ratio ($\sigma^2 \to 0$), the use of either $\hat{R}$ or $\tilde{R}$ gives exact frequency estimates, whereas the Toeplitz-averaged approximation of R does not. As $\sigma^2 \to 0$, both $\hat{R}$ and $\tilde{R}$ have rank n, but the Toeplitz-averaged approximation of R has full rank in general.

4.9 COMPLEMENTS

4.9.1 Mean-Square Convergence of Sample Covariances for Line Spectral Processes

In this complement, we prove that

$$\boxed{\lim_{N\to\infty} \hat{r}(k) = r(k) \quad \text{(in the mean-square sense)}} \tag{4.9.1}$$

(i.e., $\lim_{N\to\infty} E\left\{|\hat{r}(k) - r(k)|^2\right\} = 0$). The above result has already been referred to in Section 4.4, in the discussion on the rank properties of $\hat{\Omega}$ and Ω. It is also the basic result from which the consistency of all covariance-based frequency estimators discussed in this chapter can be readily concluded. Note that a signal $\{y(t)\}$ satisfying (4.9.1) is said to be *second-order ergodic*. (See [Söderström and Stoica 1989; Brockwell and Davis 1991] for a more detailed discussion of the ergodicity property.)

A straightforward calculation gives

$$\begin{aligned}\hat{r}(k) &= \frac{1}{N}\sum_{t=k+1}^{N}[x(t)+e(t)][x^*(t-k)+e^*(t-k)]\\ &= \frac{1}{N}\sum_{t=k+1}^{N}[x(t)x^*(t-k)+x(t)e^*(t-k)+e(t)x^*(t-k)\\ &\quad + e(t)e^*(t-k)] \triangleq T_1+T_2+T_3+T_4\end{aligned}\tag{4.9.2}$$

The limit of T_1 is found as follows. First note that

$$\begin{aligned}\lim_{N\to\infty} E\left\{|T_1 - r_x(k)|^2\right\} &= \lim_{N\to\infty}\left\{\frac{1}{N^2}\sum_{t=k+1}^{N}\sum_{s=k+1}^{N} E\left\{x(t)x^*(t-k)x^*(s)x(s-k)\right\}\right.\\ &\quad \left. -\left(\frac{2}{N}\sum_{t=k+1}^{N}|r_x(k)|^2\right)+|r_x(k)|^2\right\}\\ &= \lim_{N\to\infty}\left\{\frac{1}{N^2}\sum_{t=k+1}^{N}\sum_{s=k+1}^{N} E\left\{x(t)x^*(t-k)x^*(s)x(s-k)\right\}\right\}\\ &\quad -|r_x(k)|^2\end{aligned}$$

Now,

$$\begin{aligned}E\left\{x(t)x^*(t-k)x^*(s)x(s-k)\right\} &= \sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{m=1}^{n} a_p a_j a_l a_m e^{i(\omega_p-\omega_j)t}e^{i(\omega_m-\omega_l)s}\\ &\quad \cdot e^{i(\omega_j-\omega_m)k}E\left\{e^{i\varphi_p}e^{-i\varphi_j}e^{i\varphi_m}e^{-i\varphi_l}\right\}\\ &= \sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{m=1}^{n} a_p a_j a_l a_m e^{i(\omega_p-\omega_j)t}e^{i(\omega_m-\omega_l)s}\\ &\quad \cdot e^{i(\omega_j-\omega_m)k}\left(\delta_{p,j}\delta_{m,l}+\delta_{p,l}\delta_{m,j}-\delta_{p,j}\delta_{m,l}\delta_{p,m}\right)\end{aligned}$$

where the last equality follows from the assumed independence of the initial phases $\{\varphi_k\}$. Combining the results of the above two calculations yields

$$\begin{aligned}
\lim_{N\to\infty} E\left\{|T_1 - r_x(k)|^2\right\} &= \lim_{N\to\infty} \frac{1}{N^2} \sum_{t=k+1}^{N} \sum_{s=k+1}^{N} \left\{ \sum_{p=1}^{n}\sum_{m=1}^{n} a_p^2 a_m^2 e^{i(\omega_p-\omega_m)k} \right. \\
&\qquad \left. + \sum_{p=1}^{n}\sum_{m=1}^{n} a_p^2 a_m^2 e^{i(\omega_p-\omega_m)(t-s)} - \sum_{p=1}^{n} a_p^4 \right\} - |r_x(k)|^2 \\
&= \sum_{p=1}^{n} \sum_{\substack{m=1\\ m\neq p}}^{n} a_p^2 a_m^2 \lim_{N\to\infty} \frac{1}{N^2} \sum_{\tau=-N}^{N} (N-|\tau|) e^{i(\omega_p-\omega_m)\tau} \\
&= 0 \qquad (4.9.3)
\end{aligned}$$

It follows that T_1 converges to $r(k)$ (in the mean-square sense) as N tends to infinity.

The limits of T_2 and T_3 are equal to zero, as shown next for T_2; the proof for T_3 is similar. Using the fact that $\{x(t)\}$ and $\{e(t)\}$ are, by assumption, independent random signals, we get

$$\begin{aligned}
E\left\{|T_2|^2\right\} &= \frac{1}{N^2} \sum_{t=k+1}^{N} \sum_{s=k+1}^{N} E\left\{x(t)e^*(t-k)x^*(s)e(s-k)\right\} \\
&= \frac{\sigma^2}{N^2} \sum_{t=k+1}^{N} \sum_{s=k+1}^{N} E\left\{x(t)x^*(s)\right\} \delta_{t,s} \\
&= \frac{\sigma^2}{N^2} \sum_{t=k+1}^{N} E\left\{|x(t)|^2\right\} = \frac{(N-k)\sigma^2}{N^2} E\left\{|x(t)|^2\right\} \qquad (4.9.4)
\end{aligned}$$

which tends to zero as $N \to \infty$. Hence, T_2 (and, similarly, T_3) converges to zero in the mean-square sense.

The last term, T_4, in (4.9.2), converges to $\sigma^2\delta_{k,0}$ by the "law of large numbers" (as shown in [Söderström and Stoica 1989; Brockwell and Davis 1991]). In fact, it is readily verified, at least under the Gaussian hypothesis, that

$$\begin{aligned}
E\left\{|T_4 - \sigma^2\delta_{k,0}|^2\right\} &= \frac{1}{N^2} \sum_{t=k+1}^{N} \sum_{s=k+1}^{N} E\left\{e(t)e^*(t-k)e^*(s)e(s-k)\right\} \\
&\quad - \sigma^2\delta_{k,0} \left\{ \frac{1}{N} \sum_{t=k+1}^{N} E\left\{e(t)e^*(t-k) + e^*(t)e(t-k)\right\} \right\} \\
&\quad + \sigma^4\delta_{k,0}
\end{aligned}$$

$$= \frac{1}{N^2} \sum_{t=k+1}^{N} \sum_{s=k+1}^{N} [\sigma^4 \delta_{k,0} + \sigma^4 \delta_{t,s}]$$

$$- 2\sigma^4 \delta_{k,0} \frac{1}{N} \sum_{t=k+1}^{N} (\delta_{k,0}) + \sigma^4 \delta_{k,0}$$

$$\to \sigma^4 \delta_{k,0} - 2\sigma^4 \delta_{k,0} + \sigma^4 \delta_{k,0} = 0 \tag{4.9.5}$$

Hence, T_4 converges to $\sigma^2 \delta_{k,0}$ in the mean-square sense if $e(t)$ is Gaussian. It can be shown by using the law of large numbers that $T_4 \to \sigma^2 \delta_{k,0}$ in the mean-square sense even if $e(t)$ is non-Gaussian, as long as the fourth-order moment of $e(t)$ is finite.

Next, observe that since, for example, $E\{|T_2|^2\}$ and $E\{|T_3|^2\}$ converge to zero, then $E\{T_2 T_3^*\}$ also converges to zero (as $N \to \infty$); this is so because

$$\left| E\left\{T_2 T_3^*\right\} \right| \le \left[E\left\{|T_2|^2\right\} E\left\{|T_3|^2\right\} \right]^{1/2}$$

With this observation, the proof of (4.9.1) is complete.

4.9.2 The Carathéodory Parameterization of a Covariance Matrix

The covariance matrix model in (4.2.7) is more general than it might appear at first sight. We show that for *any* given covariance matrix $R = \{r(i-j)\}_{i,j=1}^{m}$, there exist $n \le m$, σ^2 and $\{\omega_k,\ \alpha_k\}_{k=1}^{n}$ such that R can be written as in (4.2.7). Equation (4.2.7), associated with an arbitrary given covariance matrix R, is named the *Carathéodory parameterization* of R.

Let σ^2 denote the minimum eigenvalue of R. Because σ^2 is not necessarily unique, let $\bar{n}$ denote its multiplicity and set $n = m - \bar{n}$. Define

$$\Gamma = R - \sigma^2 I$$

The matrix Γ is positive semidefinite and Toeplitz and, hence, must be the covariance matrix associated with a stationary signal, say $y(t)$:

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

By definition,

$$\operatorname{rank}(\Gamma) = n \tag{4.9.6}$$

which implies that there must exist a linear combination between $\{y(t), \dots, y(t-n)\}$ for all t. Moreover, both $y(t)$ and $y(t-n)$ must appear with nonzero coefficients in that linear combination (otherwise either $\{y(t) \dots y(t-n+1)\}$ or $\{y(t-1) \dots y(t-n)\}$ would be linearly related, and rank(Γ) would be less than n, which would contradict (4.9.6)). Hence, $y(t)$ obeys the homogeneous AR equation

$$B(z)y(t) = 0 \tag{4.9.7}$$

where z^{-1} is the unit delay operator, and

$$B(z) = 1 + b_1 z^{-1} + \cdots + b_n z^{-n}$$

with $b_n \neq 0$. Let $\phi(\omega)$ denote the PSD of $y(t)$. Then we have the following equivalences:

$$\begin{aligned} B(z)y(t) = 0 &\iff \int_{-\pi}^{\pi} |B(\omega)|^2 \, \phi(\omega) \, d\omega = 0 \\ &\iff |B(\omega)|^2 \, \phi(\omega) = 0 \\ &\iff \{\text{If } \phi(\omega) > 0 \ \text{ then } \ B(\omega) = 0\} \\ &\iff \{\phi(\omega) > 0 \text{ for at most } n \text{ values of } \omega\} \end{aligned}$$

Furthermore,

$$\begin{aligned} &\{y(t), \ldots y(t-n+1) \text{ are linearly independent}\} \\ &\iff \left\{E\left\{|g_0 y(t) + \ldots + g_{n-1} y(t-n+1)|^2\right\} > 0 \text{ for every } [g_0 \ldots g_{n-1}]^T \neq 0\right\} \\ &\iff \left\{\textstyle\int_{-\pi}^{\pi} |G(\omega)|^2 \, \phi(\omega) \, d\omega > 0 \text{ for every } G(z) = \sum_{k=0}^{n-1} g_k z^{-k} \neq 0\right\} \\ &\iff \{\phi(\omega) > 0 \text{ for at least } n \text{ distinct values of } \omega\} \end{aligned}$$

It follows from these two results that $\phi(\omega) > 0$ for exactly n distinct values of ω. Furthermore, the values of ω for which $\phi(\omega) > 0$ are given by the n roots of the equation $B(\omega) = 0$. A signal $y(t)$ with such a PSD consists of a sum of n sinusoidal components with an $m \times m$ covariance matrix given by

$$\Gamma = APA^* \tag{4.9.8}$$

(*cf.* (4.2.7)). In (4.9.8), the frequencies $\{\omega_k\}_{k=1}^{n}$ are defined as previously indicated and can be found from Γ by using any of the subspace-based frequency-estimation methods in this chapter. Once $\{\omega_k\}$ are available, $\{\alpha_i^2\}$ can be determined from Γ. (Show that.) By combining the additive decomposition $R = \Gamma + \sigma^2 I$ and (4.9.8), we obtain (4.2.7). With this observation, the derivation of the Carathéodory parameterization is complete.

It is interesting to note that the sinusoids-in-noise signal that "realizes" a given covariance sequence $\{r(0), \ldots, r(m)\}$ also provides a *positive definite extension* of that sequence. More precisely, the covariance lags $\{r(m+1), \ r(m+2), \ldots\}$ derived from the sinusoidal signal equation, when appended to $\{r(0), \ldots, r(m)\}$, provide a positive definite covariance sequence of infinite length. The AR covariance realization (see Complement 3.9.2) is the other well-known method for obtaining a positive definite extension of a given covariance sequence of finite length.

4.9.3 Using the Unwindowed Periodogram for Sine Wave Detection in White Noise

As shown in Section 4.3, the unwindowed periodogram is an accurate frequency estimation method whenever the minimum frequency separation is larger than $1/N$. A simple intuitive explanation as to why the unwindowed periodogram is a better frequency estimator than the windowed periodogram(s) follows. The principal effect of a window is to remove the tails of the sample covariance sequence from the periodogram formula; this is appropriate for signals whose covariance sequence "rapidly" goes to zero, but inappropriate for sinusoidal signals, whose covariance sequence never dies out. (For sinusoidal signals, the use of a window is expected to introduce a significant bias in the estimated spectrum.) Note, however, that, if the data contains sinusoidal components with significantly different amplitudes, then it could be advisable to use a (mildly) windowed periodogram. This will induce bias in the frequency estimates, but, on the other hand, will reduce the leakage and hence make it possible to detect the low-amplitude components.

When using the (unwindowed) periodogram for frequency estimation, an important problem is to infer whether any of the many peaks of the erratic periodogram plot can really be associated with the existence of a sinusoidal component in the data. In order to be more precise, consider the following two hypotheses:

H_0: The data consists of (complex circular Gaussian) white noise only (with unknown variance σ^2).

H_1: The data consists of a sum of sinusoidal components and noise.

Deciding between H_0 and H_1 constitutes the so-called *(signal) detection problem*. A solution to the detection problem can be obtained as follows: From the calculations leading to the result (2.4.21), one can see that the normalized periodogram values in (4.9.15) are independent random variables (under H_0). It remains to derive their distribution. Let

$$\epsilon_r(\omega) = \frac{\sqrt{2}}{\sigma\sqrt{N}} \sum_{t=1}^{N} \operatorname{Re}[e(t)e^{-i\omega t}]$$

$$\epsilon_i(\omega) = \frac{\sqrt{2}}{\sigma\sqrt{N}} \sum_{t=1}^{N} \operatorname{Im}[e(t)e^{-i\omega t}]$$

With this notation, and under the null hypothesis H_0,

$$2\hat{\phi}_p(\omega)/\sigma^2 = \epsilon_r^2(\omega) + \epsilon_i^2(\omega) \tag{4.9.9}$$

For any two complex scalars, z_1 and z_2, we have

$$\operatorname{Re}(z_1)\operatorname{Im}(z_2) = \frac{z_1 + z_1^*}{2}\,\frac{z_2 - z_2^*}{2i} = \frac{1}{2}\operatorname{Im}\,(z_1 z_2 + z_1^* z_2) \tag{4.9.10}$$

and, similarly,

$$\operatorname{Re}(z_1)\operatorname{Re}(z_2) = \frac{1}{2}\operatorname{Re}(z_1 z_2 + z_1^* z_2) \tag{4.9.11}$$

$$\operatorname{Im}(z_1)\operatorname{Im}(z_2) = \frac{1}{2}\operatorname{Re}(-z_1 z_2 + z_1^* z_2) \tag{4.9.12}$$

By making use of (4.9.10)–(4.9.12), we can write

$$\begin{aligned} E\left\{\epsilon_r(\omega)\epsilon_i(\omega)\right\} &= \frac{1}{\sigma^2 N}\operatorname{Im}\left\{\sum_{t=1}^{N}\sum_{s=1}^{N} E\left\{e(t)e(s)e^{-i\omega(t+s)} + e^*(t)e(s)e^{i\omega(t-s)}\right\}\right\} \\ &= \operatorname{Im}\{1\} = 0 \end{aligned}$$

$$\begin{aligned} E\left\{\epsilon_r^2(\omega)\right\} &= \frac{1}{\sigma^2 N}\operatorname{Re}\left\{\sum_{t=1}^{N}\sum_{s=1}^{N} E\left\{e(t)e(s)e^{-i\omega(t+s)} + e^*(t)e(s)e^{i\omega(t-s)}\right\}\right\} \\ &= \operatorname{Re}\{1\} = 1 \end{aligned} \tag{4.9.13}$$

$$\begin{aligned} E\left\{\epsilon_i^2(\omega)\right\} &= \frac{1}{\sigma^2 N}\operatorname{Re}\left\{\sum_{t=1}^{N}\sum_{s=1}^{N} E\left\{-e(t)e(s)e^{-i\omega(t+s)} + e^*(t)e(s)e^{i\omega(t-s)}\right\}\right\} \\ &= \operatorname{Re}\{1\} = 1 \end{aligned} \tag{4.9.14}$$

In addition, note that the random variables $\epsilon_r(\omega)$ and $\epsilon_i(\omega)$ are zero-mean Gaussian distributed, because they are linear transformations of the Gaussian white-noise sequence. Then, it follows that, *under* H_0,

> The random variables
>
> $$\{2\hat{\phi}_p(\omega_k)/\sigma^2\}_{k=1}^{N},$$
>
> with $\min_{k \neq j} |\omega_k - \omega_j| \geq 2\pi/N$, are asymptotically independent and χ^2 distributed with 2 degrees of freedom.

(4.9.15)

(See, e.g., [Priestley 1981] and [Söderström and Stoica 1989] for the definition and properties of the χ^2 distribution.) It is worth noting that, if $\{\omega_k\}$ are equal to the Fourier frequencies $\{2\pi k/N\}_{k=0}^{N-1}$, then the previous distributional result is *exactly valid* (i.e., it holds in samples of finite length; see, for example, equation (2.4.26)). However, this observation is not as important as it might at first seem, because σ^2 in (4.9.15) is unknown. When the noise power in (4.9.15) is replaced by a consistent estimate $\hat{\sigma}^2$, the normalized periodogram values so obtained,

$$\{2\hat{\phi}_p(\omega_k)/\hat{\sigma}^2\} \tag{4.9.16}$$

are $\chi^2(2)$ distributed only asymptotically (for $N \gg 1$). A consistent estimate of σ^2 can be obtained as follows: From (4.9.9), (4.9.13), and (4.9.14) we have that, under H_0,

$$E\left\{\hat{\phi}_p(\omega_k)\right\} = \sigma^2 \qquad \text{for } k = 1, 2, \ldots, N$$

Since $\{\hat{\phi}_p(\omega_k)\}_{k=1}^N$ are independent random variables, a consistent estimate of σ^2 is given by

$$\boxed{\hat{\sigma}^2 = \frac{1}{N} \sum_{k=1}^{N} \hat{\phi}_p\ (\omega_k)}$$

Inserting this expression for $\hat{\sigma}^2$ into (4.9.16) leads to the following "test statistic":

$$\boxed{\mu_k = \frac{2N\hat{\phi}_p(\omega_k)}{\displaystyle\sum_{k=1}^{N} \hat{\phi}_p(\omega_k)}}$$

In accordance with the (asymptotic) χ^2 distribution of $\{\mu_k\}$, we have (for any given $c \geq 0$; see, for example, [PRIESTLEY 1981])

$$\Pr(\mu_k \leq c) = \int_0^c \frac{1}{2}\, e^{-x/2}\, dx = 1 - e^{-c/2}. \tag{4.9.17}$$

Let

$$\boxed{\mu = \max_k\ [\mu_k]}$$

Using (4.9.17) (and the fact that $\{\mu_k\}$ are independent random variables) we find that (for any $c \geq 0$)

$$\begin{aligned}\Pr(\mu > c) &= 1 - \Pr(\mu \leq c)\\ &= 1 - \Pr(\mu_k \leq c \text{ for all } k)\\ &= 1 - (1 - e^{-c/2})^N \qquad \text{(under } H_0)\end{aligned}$$

This result can be used to set a bound on μ that, under H_0, holds with a (high) preassigned probability, say $1 - \alpha$. More precisely, *let α be given* (e.g., $\alpha = 0.05$), and solve for c from the equation

$$\boxed{(1 - e^{-c/2})^N = 1 - \alpha}$$

Then

- If $\mu \leq c$, accept H_0 with an unknown risk. (That risk depends on the signal-to-noise ratio (SNR). The lower the SNR, the larger the risk of accepting H_0 when it does not hold.)
- If $\mu > c$, reject H_0 with a risk equal to α.

It should be noted that, whenever H_0 is rejected by the above test, what we can really infer is that the periodogram peak in question is significant enough to make the existence of a sinusoidal component in the studied data highly probable. However, the previous test does not tell us the number of sinusoidal components in the data. In order to determine that number, the test should be continued by looking at the second-highest peak in the periodogram. For a test of the significance of the second-highest value of the periodogram, and so on, we refer to [PRIESTLEY 1981].

Finally, we note that, in addition to the test presented in this complement, there are several other tests to decide between the hypotheses H_0 and H_1; see [PRIESTLEY 1997] for a review.

4.9.4 NLS Frequency Estimation for a Sinusoidal Signal with Time-Varying Amplitude

Consider the sinusoidal data model in (4.1.1) for the case of a single component ($n = 1$), but with a time-varying amplitude

$$y(t) = \alpha(t)e^{i(\omega t+\varphi)} + e(t), \quad t = 1, \ldots, N \tag{4.9.18}$$

where $\alpha(t) \in \mathbf{R}$ is an arbitrary unknown envelope modulating the sinusoidal signal. The NLS estimates of $\alpha(t)$, ω, and φ are obtained by minimizing the criterion

$$f = \sum_{t=1}^{N} \left| y(t) - \alpha(t)e^{i(\omega t+\varphi)} \right|^2$$

(*cf.* (4.3.1)). In this complement, we show that this seemingly complicated minimization problem has, in fact, a simple solution. We also discuss briefly an FFT-based algorithm for computing that solution. The reader interested in more details on the topic of this complement can consult [BESSON AND STOICA 1999; STOICA, BESSON, AND GERSHMAN 2001] and references therein.

A straightforward calculation shows that

$$f = \sum_{t=1}^{N} \left\{ \left| y(t) \right|^2 + \left[\alpha(t) - \operatorname{Re}\left(e^{-i(\omega t+\varphi)}y(t)\right)\right]^2 - \left[\operatorname{Re}\left(e^{-i(\omega t+\varphi)}y(t)\right)\right]^2 \right\} \tag{4.9.19}$$

The minimization of (4.9.19) with respect to $\alpha(t)$ is immediate:

$$\hat{\alpha}(t) = \operatorname{Re}\left(e^{-i(\hat{\omega}t+\hat{\varphi})}y(t)\right) \tag{4.9.20}$$

Note that the NLS estimates $\hat{\omega}$ and $\hat{\varphi}$ are yet to be determined. Inserting (4.9.20) into (4.9.19) shows that the NLS estimates of φ and ω are obtained by maximizing the function

$$g = 2\sum_{t=1}^{N}\left[\operatorname{Re}\left(e^{-i(\omega t+\varphi)}y(t)\right)\right]^2$$

where the factor 2 has been introduced for the sake of convenience. For any complex number c we have

$$[\operatorname{Re}(c)]^2 = \frac{1}{4}\left(c + c^*\right)^2 = \frac{1}{2}\left[|c|^2 + \operatorname{Re}\left(c^2\right)\right]$$

It follows that

$$\begin{aligned} g &= \sum_{t=1}^{N}\left\{|y(t)|^2 + \operatorname{Re}\left[e^{-2i(\omega t+\varphi)}y^2(t)\right]\right\} \\ &= \text{constant} + \left|\sum_{t=1}^{N} y^2(t)e^{-i2\omega t}\right| \cdot \cos\left[\arg\left(\sum_{t=1}^{N} y^2(t)e^{-i2\omega t}\right) - 2\varphi\right] \end{aligned} \tag{4.9.21}$$

Clearly, the maximizing φ is given by

$$\hat{\varphi} = \frac{1}{2}\arg\left(\sum_{t=1}^{N} y^2(t)e^{-i2\hat{\omega}t}\right)$$

with the NLS estimate of ω given by

$$\boxed{\hat{\omega} = \arg\max_{\omega}\left|\sum_{t=1}^{N} y^2(t)e^{-i2\omega t}\right|} \tag{4.9.22}$$

It is important to note that the maximization in (4.9.22) should be conducted over $[0, \pi]$ instead of over $[0, 2\pi]$; indeed, the function in (4.9.22) is periodic with a period equal to π. The restriction of ω to $[0, \pi]$ is not a peculiar feature of the NLS approach; rather, it is a consequence of the generality of the problem considered in this complement. This is easily seen by making the substitution $\omega \to \omega + \pi$ in (4.9.18), which yields

$$y(t) = \tilde{\alpha}(t)e^{i(\omega t+\varphi)} + e(t), \quad t = 1, \dots, N$$

where $\tilde{\alpha}(t) = (-1)^t\alpha(t)$ is another valid (i.e., real-valued) envelope. This simple calculation confirms the fact that ω is uniquely identifiable only in the interval $[0, \pi]$. In applications, the frequency can be made to belong to $[0, \pi]$ by using a sufficiently small sampling period.

The previous estimate of ω should be contrasted with the NLS estimate of ω in the constant-amplitude case (see (4.3.11), (4.3.17)):

$$\hat{\omega} = \arg\max_{\omega} \left| \sum_{t=1}^{N} y(t) e^{-i\omega t} \right| \quad (\text{for } \alpha(t) = \text{constant}) \tag{4.9.23}$$

There is a striking similarity between (4.9.22) and (4.9.23); the only difference between these equations is the squaring of the terms in (4.9.22). As a consequence, we can apply the FFT to the squared data sequence $\{y^2(t)\}$ to obtain the $\hat{\omega}$ in (4.9.22).

The reader perhaps wonders whether there is an *intuitive* reason for the occurrence of the squared data in (4.9.22). A possible way to explain this occurrence goes as follows: Assume that $\alpha(t)$ has zero average value. Then the DFT of $\{\alpha(t)\}$, denoted $A(\bar{\omega})$, takes on small values (theoretically zero) at $\bar{\omega} = 0$. But the DFT of $\alpha(t)e^{i\omega t}$ is $A(\bar{\omega} - \omega)$, so it follows that the modulus of this DFT has a valley instead of a peak at $\bar{\omega} = \omega$; hence, the standard periodogram (see (4.9.23)) should not be used to estimate ω. On the other hand, $\alpha^2(t)$ always has a nonzero average value (or DC component); hence, the modulus of the DFT of $\alpha^2(t)e^{i2\omega t}$ will typically have a peak at $\bar{\omega} = 2\omega$. This observation provides an heuristic reason for the squaring operation in (4.9.22).

4.9.5 Monotonically Descending Techniques for Function Minimization

As was explained in Section 4.3, minimizing the NLS criterion with respect to the unknown frequencies is made rather difficult by existence of possibly many local minima and by the sharpness of the global minimum. In this complement (based on [Stoica and Selén 2004a]), we will discuss a number of methods that can be used to solve such a minimization problem. Our discussion is quite general and applies to many other functions, not to just the NLS criterion that is used as an illustrating example in what follows.

We will denote the function to be minimized by $f(\theta)$, where θ is a vector. Sometimes we will write this function as $f(x, y)$ where $[x^T, y^T]^T = \theta$. The algorithms for minimizing $f(\theta)$ discussed in this complement are iterative. We let θ^i denote the value taken by θ at the ith iteration (and similarly for x and y). The *common feature* of the algorithms included in this complement is that *they all monotonically decrease the function at each iteration*:

$$\boxed{f(\theta^{i+1}) \leq f(\theta^i) \quad \text{for } i = 0, 1, 2, \ldots} \tag{4.9.24}$$

Hereafter, θ^0 denotes the initial value (or estimate) of θ used by the minimization algorithm in question. Clearly, (4.9.24) is an appealing property, which is the main reason for the interest in the algorithms discussed here. However, we should note that usually (4.9.24) can do no more than guarantee the convergence to a *local minimum* of $f(\theta)$. The goodness of the initial estimate θ^0 will often determine whether the algorithm will converge to the global minimum. In fact, for some of the algorithms to be discussed, not even the convergence to a local minimum is guaranteed. For example, the EM algorithm (discussed later in this complement) can converge to saddle points or local maxima. (See, for example, [McLachlan and Krishnan 1997].) However, such a behavior is rare in applications, provided that some regularity conditions are satisfied.

Cyclic Minimizer

To describe the main idea of this type of algorithm in its simplest form, let us partition θ into two subvectors:

$$\theta = \begin{bmatrix} x \\ y \end{bmatrix}$$

Then the *generic iteration of a cyclic algorithm* for minimizing $f(x, y)$ will have the following form:

$$\boxed{\begin{aligned} &y^0 = \text{ given} \\ &\text{For } i = 1, 2, \ldots \text{ compute:} \\ &\qquad x^i = \arg\min_x f(x, y^{i-1}) \\ &\qquad y^i = \arg\min_y f(x^i, y) \end{aligned}} \tag{4.9.25}$$

Note that (4.9.25) alternates (or cycles) between the minimization of $f(x, y)$ with respect to x for given y and the minimization of $f(x, y)$ with respect to y for given x; hence, the name "cyclic" given to this type of algorithm. An obvious modification of (4.9.25) allows us to start with x^0, if so desired. It is readily verified that the cyclic minimizer (4.9.25) possesses the property (4.9.24)—that is,

$$f(x^i, y^i) \leq f(x^i, y^{i-1}) \leq f(x^{i-1}, y^{i-1})$$

where the first inequality follows from the definition of y^i and the second from the definition of x^i.

The partitioning of θ into subvectors is usually done in such a way that the minimization operations in (4.9.25) *(or at least one of them) are "easy"* (in any case, easier than the minimization of f jointly with respect to x and y). Quite often, to achieve this desired property, we need to partition θ into more than two subvectors. The extension of (4.9.25) to such a case is straightforward and will not be discussed here. However, there is one point about this extension that we would like to make briefly: whenever θ is partitioned into three or more subvectors, we can choose the way in which the various minimization subproblems are iterated. For instance, if $\theta = [x^T, y^T, z^T]^T$ then we may iterate the minimization steps with respect to x and with respect to y a number of times (with z being fixed), before reestimating z, and so forth.

With reference to the NLS problem in Section 4.3, we can apply the preceding ideas to the following natural partitioning of the parameter vector:

$$\theta = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix}, \qquad \gamma_k = \begin{bmatrix} \omega_k \\ \varphi_k \\ \alpha_k \end{bmatrix} \tag{4.9.26}$$

The main virtue of this partitioning of θ is that the problem of minimizing the NLS criterion with respect to γ_k, for given $\{\gamma_j\}$ $(j = 1, \ldots, n; j \neq k)$, can be solved via the FFT (see (4.3.10), (4.3.11)). Furthermore, the cyclic minimizer corresponding to (4.9.26) can be initialized, with $\gamma_2 = \cdots = \gamma_n = 0$, in which case the γ_1 minimizing the NLS criterion is obtained from the highest peak of the periodogram (which should give a reasonably accurate estimate of γ_1), and so on.

An elaborated cyclic algorithm, called RELAX, for the minimization of the NLS criterion based on the preceding ideas (see (4.9.26)), was proposed in [Li and Stoica 1996b]. Note that cyclic minimizers are sometimes called *relaxation algorithms*, which provide a motivation for the name given to the algorithm in [Li and Stoica 1996b].

Majorization Technique

The main idea of this type of iterative technique for minimizing a given function $f(\theta)$ is quite simple. (See, for example, [Heiser 1995] and the references therein.) Assume that, at the ith iteration, we can find a function $g_i(\theta)$ (the subindex i indicates the dependence of this function on θ^i) that possesses the following three properties:

$$g_i(\theta^i) = f(\theta^i) \tag{4.9.27}$$

$$g_i(\theta) \geq f(\theta) \tag{4.9.28}$$

and

the minimization of $g_i(\theta)$ with respect to θ is "easy" (or, in any case, easier than the minimization of $f(\theta)$). (4.9.29)

Owing to (4.9.28), $g_i(\theta)$ is called a *majorizing function* for $f(\theta)$ at the ith iteration. In the majorization technique, the parameter vector at iteration $(i + 1)$ is obtained from the minimization of $g_i(\theta)$:

$$\boxed{\theta^{i+1} = \arg\min_{\theta} g_i(\theta)} \tag{4.9.30}$$

The key property (4.9.24) is satisfied for (4.9.30), since

$$f(\theta^i) = g_i(\theta^i) \geq g_i(\theta^{i+1}) \geq f(\theta^{i+1}) \tag{4.9.31}$$

The first inequality in (4.9.31) follows from the definition of θ^{i+1} in (4.9.30), the second from (4.9.28). Note that, in fact, from (4.9.31), we get

$$f(\theta^i) - f(\theta^{i+1}) \geq g_i(\theta^i) - g_i(\theta^{i+1}) \geq 0$$

which shows not only that $f(\theta)$ is monotonically decreased at each iteration but also that the decrease in $f(\theta)$ is not smaller than the corresponding decrease of the majorizing function $g_i(\theta)$.

Note that any parameter vector θ^{i+1} that gives a smaller value of $g_i(\theta)$ than does $g_i(\theta^i)$ will satisfy (4.9.31). Consequently, whenever the minimum point of $g_i(\theta)$ (see (4.9.30)) cannot be derived in closed form, we can think of computing θ^{i+1} by, for example, performing a few iterations with a gradient-based algorithm initialized at θ^i and using a line search (to guarantee that $g_i(\theta^{i+1}) \leq g_i(\theta^i)$). A similar observation could be made on the cyclic minimizer in (4.9.25) when the minimization of either $f(x, y^{i-1})$ or $f(x^i, y)$ cannot be done in closed form. The modification of either (4.9.30) or (4.9.25) in this way usually simplifies the computational effort of each iteration, but could slow down the convergence speed of the algorithm by increasing the number of iterations needed to achieve convergence.

An interesting question regarding the two algorithms discussed so far is whether we could obtain the cyclic minimizer by using the majorization principle on a certain majorizing function. In general, it appears difficult or impossible to do so; nor can the majorization technique be obtained as a special case of a cyclic minimizer. Hence, these two iterative minimization techniques appear to have “independent lives.”

To draw more parallels between the cyclic minimizer and the majorization technique, we remark on the fact that, in the former, the user has to choose the partitioning of θ that makes the minimization in, for example, (4.9.25) “easy,” whereas in the latter a function $g_i(\theta)$ has to be found that is not only “easy” to minimize but also possesses the essential property (4.9.28). Fortunately for the majorization approach, finding such functions $g_i(\theta)$ is not as hard as it might at first seem. In what follows, we will develop a method for constructing a function $g_i(\theta)$ possessing the desired properties (4.9.27) and (4.9.28) for a *general class* of functions $f(\theta)$ (including the NLS criterion) that are commonly encountered in parameter estimation applications.

EM Algorithm

The NLS criterion (see (4.3.1)),

$$f(\theta) = \sum_{t=1}^{N} \left| y(t) - \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \varphi_k)} \right|^2 \tag{4.9.32}$$

where θ is defined in (4.9.26), is obtained from the data equation (4.1.1) in which the noise $\{e(t)\}$ is assumed to be circular and white with mean zero and variance σ^2. Let us also assume that $\{e(t)\}$ is Gaussian distributed; then the probability density function of the data vector $y = [y(1), \ldots, y(N)]^T$, for given θ, is

$$p(y, \theta) = \frac{1}{(\pi\sigma^2)^N} e^{-\frac{f(\theta)}{\sigma^2}} \tag{4.9.33}$$

where $f(\theta)$ is as defined in (4.9.32). The *method of maximum likelihood* (ML) obtains an estimate of θ by maximizing (4.9.33) (see (B.1.7) in Appendix B) or, equivalently, by minimizing the so-called *negative log-likelihood function*:

$$-\ln p(y, \theta) = \text{constant} + N \ln \sigma^2 + \frac{f(\theta)}{\sigma^2} \tag{4.9.34}$$

Minimizing (4.9.34) with respect to θ is equivalent to minimizing (4.9.32), which shows that the NLS method is identical to the ML method under the assumption that $\{e(t)\}$ is Gaussian white noise.

The ML is without a doubt the most widely studied method of parameter estimation. In what follows, we assume that this is the method used for parameter estimation and hence that the function we want to minimize with respect to θ is the negative log-likelihood:

$$\boxed{f(\theta) = -\ln p(y, \theta)} \tag{4.9.35}$$

Our main goal in this subsection is to show how to construct *a majorizing function for the estimation criterion* in (4.9.35) and how the use of *the corresponding majorization technique leads to the expectation-maximization (EM) algorithm* introduced in [Dempster, Laird, and Rubin 1977]. See also [McLachlan and Krishnan 1997] and [Moon 1996] for more recent and detailed accounts on the EM algorithm.

A notation that will be frequently used concerns the expectation with respect to the distribution of a certain random vector—say z—which we will denote by $E_z\{\cdot\}$. When the distribution concerned is conditioned on another random vector—say y—we will use the notation $E_{z|y}\{\cdot\}$. If we also want to stress the dependence of the distribution (with respect to which the expectation is taken) on a certain parameter vector θ, then we write $E_{z|y,\theta}\{\cdot\}$.

The main result that we will use in the following is *Jensen's inequality*. It asserts that, for any *concave function* $h(x)$, where x is a random vector, the following inequality holds:

$$\boxed{E\{h(x)\} \leq h(E\{x\})} \tag{4.9.36}$$

The proof of (4.9.36) is simple. Let $d(x)$ denote the plane tangent to $h(x)$ at the point $E\{x\}$. Then

$$E\{h(x)\} \leq E\{d(x)\} = d(E\{x\}) = h(E\{x\}) \tag{4.9.37}$$

which proves (4.9.36). The inequality in (4.9.37) follows from the concavity of $h(x)$, the first equality follows from the fact that $d(x)$ is a linear function of x, and the second equality from the fact that $d(x)$ is tangent (and hence equal) to $h(x)$ at the point $E\{x\}$.

Remark: We note in passing that, despite its simplicity, Jensen's inequality is a powerful analysis tool. As a simple illustration of this fact, consider a scalar random variable x with a discrete probability distribution:

$$\Pr\{x = x_k\} = p_k, \quad k = 1, \ldots, M$$

Then, using (4.9.36) and the fact that the *logarithm is a concave function*, we obtain (assuming $x_k > 0$)

$$E\{\ln(x)\} = \sum_{k=1}^{M} p_k \ln(x_k) \leq \ln\left[E\{x\}\right] = \ln\left[\sum_{k=1}^{M} p_k x_k\right]$$

or, equivalently,

$$\sum_{k=1}^{M} p_k x_k \geq \prod_{k=1}^{M} x_k^{p_k} \quad \left(\text{for } x_k > 0 \text{ and } \sum_{k=1}^{M} p_k = 1\right) \tag{4.9.38}$$

For $p_k = 1/M$, (4.9.38) reduces to the well-known inequality between the arithmetic and geometric means—that is,

$$\frac{1}{M}\sum_{k=1}^{M} x_k \geq \left(\prod_{k=1}^{M} x_k\right)^{1/M}$$

which is so easily obtained in the present framework. ■

After these preparations, we turn our attention to the main question of finding a majorizing function for (4.9.35). Let z be a random vector whose probability density function conditioned on y is completely determined by θ, and let

$$g_i(\theta) = f(\theta^i) - E_{z|y,\theta^i}\left\{\ln\left[\frac{p(y,z,\theta)}{p(y,z,\theta^i)}\right]\right\} \tag{4.9.39}$$

Clearly $g_i(\theta)$ satisfies

$$g_i(\theta^i) = f(\theta^i) \tag{4.9.40}$$

Furthermore, it follows from Jensen's inequality (4.9.36), the concavity of the function $\ln(\cdot)$, and Bayes' rule for conditional probabilities that

$$\begin{aligned} g_i(\theta) &\geq f(\theta^i) - \ln\left[E_{z|y,\theta^i}\left\{\frac{p(y,z,\theta)}{p(y,z,\theta^i)}\right\}\right] \\ &= f(\theta^i) - \ln\left[E_{z|y,\theta^i}\left\{\frac{p(y,z,\theta)}{p(z|y,\theta^i)p(y,\theta^i)}\right\}\right] \\ &= f(\theta^i) - \ln\left[\frac{1}{p(y,\theta^i)}\underbrace{\int p(y,z,\theta)\,dz}_{p(y,\theta)}\right] \\ &= f(\theta^i) - \ln\left[\frac{p(y,\theta)}{p(y,\theta^i)}\right] \\ &= f(\theta^i) + \left[f(\theta) - f(\theta^i)\right] = f(\theta) \end{aligned} \tag{4.9.41}$$

which shows that the function $g_i(\theta)$ in (4.9.39) also satisfies the key majorization condition (4.9.28). Usually, z is called the *unobserved data* (to distinguish it from the observed data vector y), and the combination (z, y) is called the *complete data*, while y is called the *incomplete data*.

It follows from (4.9.40) and (4.9.41), along with the discussion in the previous subsection about the majorization approach, that the following algorithm *will monotonically reduce the negative log-likelihood function at each iteration*:

The Expectation–Maximization (EM) Algorithm

```
$\theta^0 =$ given
For $i = 0, 1, 2, \ldots$:                                                    (4.9.42)
    Expectation step: Evaluate $E_{z|y,\theta^i}\{\ln p(y, z, \theta)\} \triangleq \overline{g}_i(\theta)$
    Maximization step: Compute $\theta^{i+1} = \arg\max_{\theta} \overline{g}_i(\theta)$
```

This is the *EM algorithm* in a nutshell.

An important aspect of the EM algorithm, which must be considered in every application, is *the choice of the unobserved data vector z*. This choice should be done such that the maximization step of (4.9.42) is “easy” or, in any case, much easier than the maximization of the likelihood function. In general, doing so is not an easy task. In addition, the evaluation of the conditional expectation in (4.9.42) might also be rather challenging. Somewhat paradoxically, these difficulties associated with the EM algorithm have perhaps been a cause for its considerable popularity. Indeed, the detailed derivation of the EM algorithm for a particular application is a more challenging research problem (and hence more appealing to many researchers) than, for instance, the derivation of a cyclic minimizer (which also possesses the key property (4.9.24) of the EM algorithm).

4.9.6 Frequency-Selective ESPRIT-Based Method

In several applications of spectral analysis, the user is interested only in the components lying in a small frequency band of the spectrum. A *frequency-selective* method deals precisely with this kind of spectral analysis: It estimates the parameters of only those sinusoidal components in the data that lie in a prespecified band of the spectrum, with as little interference as possible from the out-of-band components, and in a computationally efficient way. To be more specific, let us consider the sinusoidal data model in (4.1.1):

$$y(t) = \sum_{k=1}^{\bar{n}} \beta_k e^{i\omega_k t} + e(t); \quad \beta_k = \alpha_k e^{i\varphi_k}, \quad t = 0, \ldots, N-1 \tag{4.9.43}$$

In some applications (see, e.g., [McKelvey and Viberg 2001; Stoica, Sandgren, Selén, Vanhamme, and Van Huffel 2003] and the references therein), it would be computationally too intensive to estimate the parameters of all components in (4.9.43). For instance, this is the case when $\bar{n}$ takes on values close to N or when $\bar{n} \ll N$ but we have many sets of data to process.

In such applications, because of computational and other reasons (see points (i) and (ii) below for details), we focus on only those components of (4.9.43) that are of direct interest to us. Let us assume that the components of interest lie in a prespecified frequency band composed of the following Fourier frequencies:

$$\left\{ \frac{2\pi}{N}k_1, \frac{2\pi}{N}k_2, \ldots, \frac{2\pi}{N}k_M \right\} \tag{4.9.44}$$

where $\{k_1, \ldots, k_M\}$ are M given (typically consecutive) integers. We assume that the number of components of (4.9.43) lying in (4.9.44), which we denote by

$$n \leq \bar{n} \tag{4.9.45}$$

is given. If n is *a priori* unknown, then it could be estimated from the data by the methods described in Appendix C.

Our problem is to estimate the parameters of the n components of (4.9.43) that lie in the frequency band in (4.9.44). Furthermore, we want to find a solution to this frequency-selective estimation problem that has the following properties:

(i) *It is computationally efficient.* In particular, the computational complexity of such a solution should be comparable with that of a standard ESPRIT method for a sinusoidal model with n components.

(ii) *It is statistically accurate.* To be more specific about this aspect, we will split the discussion into two parts. From a theoretical standpoint, estimating $n < \bar{n}$ components of (4.9.43) (in the presence of the remaining components and noise) cannot produce more accurate estimates than estimating all $\bar{n}$ components. However, for a good frequency-selective method, the degradation of theoretical statistical accuracy should not be significant. On the other hand, from a practical standpoint, a sound frequency-selective method could give better performance than a non-frequency-selective counterpart that deals with all $\bar{n}$ components of (4.9.43). This is so because some components of (4.9.43) that do not belong to (4.9.44) might not be well-described by a sinusoidal model; consequently, treating such components as interference and eliminating them from the model could improve the estimation accuracy of the components of interest.

In this complement, following [McKelvey and Viberg 2001] and [Stoica, Sandgren, Selén, Vanhamme, and Van Huffel 2003], we present a *frequency-selective ESPRIT-based* (FRES-ESPRIT) method that possesses the previous two desirable features. The following notation will be used frequently in what follows:

$$w_k = e^{i\frac{2\pi}{N}k} \quad k = 0, 1, \ldots, N-1 \tag{4.9.46}$$

$$u_k = [w_k, \ldots, w_k^m]^T \tag{4.9.47}$$

$$v_k = [1, w_k, \ldots, w_k^{N-1}]^T \tag{4.9.48}$$

$$y = [y(0), \ldots, y(N-1)]^T \tag{4.9.49}$$

$$Y_k = v_k^* y \qquad k = 0, 1, \ldots, N-1 \tag{4.9.50}$$

$$e = [e(0), \ldots, e(N-1)]^T \tag{4.9.51}$$

$$E_k = v_k^* e \qquad k = 0, 1, \ldots, N-1 \tag{4.9.52}$$

$$a(\omega_k) = \left[e^{i\omega_k}, \ldots, e^{im\omega_k}\right]^T \tag{4.9.53}$$

$$b(\omega_k) = \left[1, e^{i\omega_k}, \ldots, e^{i(N-1)\omega_k}\right]^T \tag{4.9.54}$$

Hereafter, m is a *user parameter* whose choice will be discussed later on. Note that $\{Y_k\}$ is *the FFT of the data.*

First, we show that the following key equation involving the FFT sequence $\{Y_k\}$ holds true:

$$u_k Y_k = [a(\omega_1), \ldots, a(\omega_{\bar{n}})] \begin{bmatrix} \beta_1 v_k^* b(\omega_1) \\ \vdots \\ \beta_{\bar{n}} v_k^* b(\omega_{\bar{n}}) \end{bmatrix} + \Gamma u_k + u_k E_k \tag{4.9.55}$$

Here Γ is an $m \times m$ matrix defined in equation (4.9.61). (It will become clear shortly that the definition of Γ has no importance for what follows; hence, it is not repeated here.)

To prove (4.9.55), we first write the data vector y as

$$y = \sum_{\ell=1}^{\bar{n}} \beta_\ell b(\omega_\ell) + e \tag{4.9.56}$$

Next, we note that (for $p = 1, \ldots, m$)

$$\begin{aligned}
w_k^p \left[v_k^* b(\omega)\right] &= \sum_{t=0}^{N-1} e^{i\left(\omega - \frac{2\pi}{N}k\right)t} e^{i\frac{2\pi}{N}kp} \\
&= e^{i\omega p} \sum_{t=0}^{N-1} e^{i\left(\omega - \frac{2\pi}{N}k\right)(t-p)} \\
&= e^{i\omega p}\left[v_k^* b(\omega)\right] + e^{i\omega p}\left[\sum_{t=0}^{p-1} e^{i\omega(t-p)} e^{-i\frac{2\pi}{N}k(t-p)} \right. \\
&\qquad\qquad \left. - \sum_{t=N}^{N+p-1} e^{i\omega(t-p)} e^{-i\frac{2\pi}{N}k(t-p)}\right] \\
&= e^{i\omega p}\left[v_k^* b(\omega)\right] + e^{i\omega p} \sum_{\ell=1}^{p} \left[e^{-i\omega\ell} e^{i\frac{2\pi}{N}k\ell} - e^{i\omega(N-\ell)} e^{i\frac{2\pi}{N}k\ell}\right] \\
&= e^{i\omega p}\left[v_k^* b(\omega)\right] + \sum_{\ell=1}^{p} e^{i\omega(p-\ell)} \left(1 - e^{i\omega N}\right) w_k^\ell
\end{aligned} \tag{4.9.57}$$

Let (for $p = 1, \ldots, m$)

$$\gamma_p^*(\omega) = \left(1 - e^{i\omega N}\right)\left[e^{i\omega(p-1)}, e^{i\omega(p-2)}, \ldots, e^{i\omega}, 1, 0, \ldots, 0\right] \quad (1 \times m) \tag{4.9.58}$$

Using (4.9.58), we can rewrite (4.9.57) in the following more compact form (for $p = 1, \ldots, m$):

$$w_k^p \left[v_k^* b(\omega)\right] = e^{i\omega p}\left[v_k^* b(\omega)\right] + \gamma_p^*(\omega) u_k \tag{4.9.59}$$

or, equivalently,

$$u_k \left[v_k^* b(\omega)\right] = a(\omega)\left[v_k^* b(\omega)\right] + \begin{bmatrix} \gamma_1^*(\omega) \\ \vdots \\ \gamma_m^*(\omega) \end{bmatrix} u_k \tag{4.9.60}$$

From (4.9.56) and (4.9.60), it follows that

$$\begin{aligned} u_k Y_k &= \sum_{\ell=1}^{\bar{n}} \beta_\ell u_k \left[v_k^* b(\omega_\ell)\right] + u_k E_k \\ &= [a(\omega_1), \ldots, a(\omega_{\bar{n}})] \begin{bmatrix} \beta_1 v_k^* b(\omega_1) \\ \vdots \\ \beta_{\bar{n}} v_k^* b(\omega_{\bar{n}}) \end{bmatrix} + \left\{ \sum_{\ell=1}^{\bar{n}} \beta_\ell \begin{bmatrix} \gamma_1^*(\omega_\ell) \\ \vdots \\ \gamma_m^*(\omega_\ell) \end{bmatrix} \right\} u_k + u_k E_k \end{aligned} \tag{4.9.61}$$

which proves (4.9.55).

Next, we let $\{\omega_k\}_{k=1}^n$ denote *the frequencies of interest* (i.e., those frequencies of (4.9.43) that lie in (4.9.44)). To separate the terms in (4.9.55) corresponding to the components of interest from those associated with the nuisance components, we use the notation

$$A = [a(\omega_1), \ldots, a(\omega_n)] \tag{4.9.62}$$

$$x_k = \begin{bmatrix} \beta_1 v_k^* b(\omega_1) \\ \vdots \\ \beta_n v_k^* b(\omega_n) \end{bmatrix} \tag{4.9.63}$$

for the components of interest, and similarly $\tilde{A}$ and $\tilde{x}_k$ for the other components. Finally, to write the equation (4.9.55) for $k = k_1, \ldots, k_M$ in a compact matrix form, we need the additional notation

$$Y = \left[u_{k_1} Y_{k_1}, \ldots, u_{k_M} Y_{k_M}\right], \qquad (m \times M) \tag{4.9.64}$$

$$E = \left[u_{k_1} E_{k_1}, \ldots, u_{k_M} E_{k_M}\right], \qquad (m \times M) \tag{4.9.65}$$

$$U = \left[u_{k_1}, \ldots, u_{k_M}\right], \qquad (m \times M) \tag{4.9.66}$$

$$X = \left[x_{k_1}, \ldots, x_{k_M}\right], \qquad (n \times M) \tag{4.9.67}$$

and similarly for $\tilde{X}$. Using this notation, we can write (4.9.55) (for $k = k_1, \ldots, k_M$) as follows:

$$Y = AX + \Gamma U + \tilde{A}\tilde{X} + E \tag{4.9.68}$$

Next, we assume that

$$\boxed{M \geq n + m} \tag{4.9.69}$$

which can be satisfied by choosing the user parameter m appropriately. Under (4.9.69) (in fact only $M \geq m$ is required for this part), the orthogonal projection matrix onto the null space of U is given by (see Appendix A)

$$\Pi_U^{\perp} = I - U^* \left(UU^*\right)^{-1} U \tag{4.9.70}$$

We will eliminate the second term in (4.9.68) by postmultiplying (4.9.68) with $\Pi_U^{\perp}$. However, before doing so, we make the following observations about the third and fourth terms in (4.9.68):

(a) The elements of the noise term E in (4.9.68) are much smaller than the elements of AX. In effect, it can be shown that $E_k = \mathcal{O}\left(N^{1/2}\right)$ (stochastically), whereas the order of the elements of X is typically $\mathcal{O}(N)$.

(b) Assuming that the out-of-band components are not much stronger than the components of interest, and that the frequencies of the former are not too close to the interval of interest in (4.9.44), the elements of $\tilde{X}$ are also much smaller than the elements of X.

(c) To understand what happens in the case that the assumption made in (b) does not hold, let us consider a generic out-of-band component (ω, β). The part of y corresponding to this component can be written as $\beta b(\omega)$. Hence, the corresponding part in $u_k Y_k$ is given by $\beta u_k \left[v_k^* b(\omega)\right]$; consequently, the part of Y due to this generic component is

$$\beta U \begin{bmatrix} v_{k_1}^* b(\omega) & & 0 \\ & \ddots & \\ 0 & & v_{k_M}^* b(\omega) \end{bmatrix} \tag{4.9.71}$$

Even if ω is relatively close to the band of interest, (4.9.44), we may expect that $v_k^* b(\omega)$ does not vary significantly for $k \in [k_1, k_M]$ (in other words, the "spectral tail" of the out-of-band component could well have a small dynamic range in the interval of interest). As a consequence, the matrix in (4.9.71) will be approximately proportional to U and hence it will be attenuated via the postmultiplication of it by $\Pi_U^{\perp}$ (see below). A similar argument shows that the noise term in (4.9.68) is also attenuated by postmultiplying (4.9.68) with $\Pi_U^{\perp}$.

It follows from the previous discussion and (4.9.68) that

$$Y \Pi_U^{\perp} \simeq AX \Pi_U^{\perp} \tag{4.9.72}$$

This equation resembles equation (4.7.7), on which the standard ESPRIT method is based, provided that

$$\operatorname{rank}\left(X\Pi_U^{\perp}\right) = n \tag{4.9.73}$$

(similarly to rank$(C) = n$ for (4.7.7)). In the following, we prove that (4.9.73) *holds under* (4.9.69) *and the regularity condition that* $e^{iN\omega_k} \neq 1$ (for $k = 1, \dots, n$).

To prove (4.9.73), we first note that $\operatorname{rank}\left(\Pi_U^{\perp}\right) = M - m$, which implies that $M \geq m + n$ (i.e., (4.9.69)) is a necessary condition for (4.9.73) to hold.

Next we show that (4.9.73) is equivalent to

$$\operatorname{rank}\left(\begin{bmatrix} X \\ U \end{bmatrix}\right) = m + n \tag{4.9.74}$$

To verify this equivalence, let us decompose X additively as

$$X = X\Pi_U + X\Pi_U^{\perp} = XU^*\left(UU^*\right)^{-1}U + XV^*V \tag{4.9.75}$$

where the $M \times (M - m)$ matrix V^* comprises a unitary basis of $\mathcal{N}(U)$; hence, $UV^* = 0$ and $VV^* = I$. Now, the matrix in (4.9.74) has the same rank as

$$\begin{bmatrix} I & -XU^*(UU^*)^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} XV^*V \\ U \end{bmatrix} \tag{4.9.76}$$

(we used (4.9.75) to obtain (4.9.76)), which, in turn, has the same rank as

$$\begin{bmatrix} XV^*V \\ U \end{bmatrix}\begin{bmatrix} V^*VX^* & U^* \end{bmatrix} = \begin{bmatrix} XV^*VX^* & 0 \\ 0 & UU^* \end{bmatrix} \tag{4.9.77}$$

However, rank$(UU^*) = m$. Thus, (4.9.74) holds if and only if

$$\operatorname{rank}(XV^*VX^*) = n$$

As

$$\operatorname{rank}(XV^*VX^*) = \operatorname{rank}(X\Pi_U^{\perp}X^*) = \operatorname{rank}(X\Pi_U^{\perp})$$

the equivalence between (4.9.73) and (4.9.74) is proven.

It follows from this equivalence and the definition of X and U that we want to prove that

$$\operatorname{rank}\left\{\underbrace{\begin{bmatrix} v_{k_1}^* b(\omega_1) & \cdots & v_{k_M}^* b(\omega_1) \\ \vdots & & \vdots \\ v_{k_1}^* b(\omega_n) & \cdots & v_{k_M}^* b(\omega_n) \\ u_{k_1} & \cdots & u_{k_M} \end{bmatrix}}_{(n+m)\times M}\right\} = n + m \tag{4.9.78}$$

Now,

$$v_k^* b(\omega) = \sum_{t=0}^{N-1} e^{i\left(\omega - \frac{2\pi}{N}k\right)t} = \frac{1 - e^{iN\left(\omega - \frac{2\pi}{N}k\right)}}{1 - e^{i\left(\omega - \frac{2\pi}{N}k\right)}} = \frac{1 - e^{iN\omega}}{w_k - e^{i\omega}} w_k$$

so we can rewrite the matrix in (4.9.78) as follows:

$$\begin{bmatrix} 1 - e^{iN\omega_1} & & & & & 0 \\ & \ddots & & & & \\ & & 1 - e^{iN\omega_n} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ 0 & & & & & 1 \end{bmatrix} \begin{bmatrix} \frac{w_{k_1}}{w_{k_1} - e^{i\omega_1}} & \cdots & \frac{w_{k_M}}{w_{k_M} - e^{i\omega_1}} \\ \vdots & & \vdots \\ \frac{w_{k_1}}{w_{k_1} - e^{i\omega_n}} & \cdots & \frac{w_{k_M}}{w_{k_M} - e^{i\omega_n}} \\ w_{k_1} & \cdots & w_{k_M} \\ \vdots & & \vdots \\ w_{k_1}^m & \cdots & w_{k_M}^m \end{bmatrix} \tag{4.9.79}$$

Because, by assumption, $1 - e^{iN\omega_k} \neq 0$ (for $k = 1, \ldots, n$), it follows that (4.9.78) holds if and only if the second matrix in (4.9.79) has full row rank (under (4.9.69)), which holds true if and only if we cannot find some numbers $\{\rho_k\}_{k=1}^{m+n}$ (not all zero) such that

$$\begin{aligned} &\frac{\rho_1 z}{z - e^{i\omega_1}} + \cdots + \frac{\rho_n z}{z - e^{i\omega_n}} + \rho_{n+1} z + \cdots + \rho_{n+m} z^m \\ &\qquad = z\left(\frac{\rho_1}{z - e^{i\omega_1}} + \cdots + \frac{\rho_n}{z - e^{i\omega_n}} + \rho_{n+1} + \cdots + \rho_{n+m} z^{m-1}\right) \end{aligned} \tag{4.9.80}$$

is equal to zero at $z = w_{k_1}, \ldots, z = w_{k_M}$. However, (4.9.80) can have at most $m + n - 1$ zeroes of this form, and $m + n - 1 < M$ from (4.9.69). With this observation, the proof of (4.9.73) is concluded.

To make use of (4.9.72) and (4.9.73) in an ESPRIT-like approach, we also assume that

$$\boxed{m \geq n} \tag{4.9.81}$$

(which is an easily satisfied condition); then it follows from (4.9.72) and (4.9.73) that the effective rank of the "data" matrix $Y\Pi_U^\perp$ is n, and that

$$\boxed{\hat{S} \simeq A\hat{C}} \tag{4.9.82}$$

where $\hat{C}$ is an $n \times n$ nonsingular transformation matrix, and

$$\boxed{\begin{array}{l}\hat{S} = \text{the } m \times n \text{ matrix whose columns are the left singular vectors of } Y\Pi_U^\perp \\ \text{associated with the } n \text{ largest singular values.}\end{array}} \tag{4.9.83}$$

Equation (4.9.82) is very similar to (4.7.7); hence, it can be used in *an ESPRIT-like approach to estimate the frequencies* $\{\omega_k\}_{k=1}^n$. After the frequency-estimation step, the amplitudes $\{\beta_k\}_{k=1}^n$ can be estimated, for instance, as described in [McKelvey and Viberg 2001; Stoica, Sandgren, Selén, Vanhamme, and Van Huffel 2003].

An implementation detail that we would like to address, at least briefly, is the choice of m. We recommend choosing m as the integer part of $M/2$; that is,

$$\boxed{m = \lfloor M/2 \rfloor} \tag{4.9.84}$$

provided that $\lfloor M/2 \rfloor \in [n, M-n]$, to satisfy the assumptions in (4.9.69) and (4.9.81). To motivate this choice of m, we refer to the matrix equation (4.9.72) that lies at the basis of the proposed estimation approach. Previous experience with ESPRIT, MUSIC, and other, similar approaches has shown that their accuracy increases as the number of *independent* equations in (4.9.72) (and its counterparts) increases. The matrix $Y\Pi_U^\perp$ in (4.9.72) is $m \times M$, and its rank is generically equal to

$$\min\{\text{rank}(Y),\ \text{rank}(\Pi_U^\perp)\} = \min(m, M-m) \tag{4.9.85}$$

Evidently, this rank determines the aforementioned number of linearly independent equations in (4.9.72). Hence, for enhanced estimation accuracy, we should maximize (4.9.85) with respect to m: the solution is clearly given by (4.9.84).

To end this complement, we show that *the proposed FRES-ESPRIT method with $M = N$ is equivalent to the standard ESPRIT method.* For $M = N$, we have that

$$[b_1, \ldots, b_N] \triangleq \begin{bmatrix} w_1 & \cdots & w_N \\ w_1^2 & \cdots & w_N^2 \\ \vdots & & \vdots \\ w_1^N & \cdots & w_N^N \end{bmatrix} = \underbrace{\begin{bmatrix} U \\ \bar{U} \end{bmatrix}}_{N} \begin{array}{l} \}\, m \\ \}\, N-m \end{array} \tag{4.9.86}$$

where U is as defined before (with $M = N$) and $\bar{U}$ is defined via (4.9.86). Note that

$$UU^* = NI; \qquad \bar{U}\bar{U}^* = NI; \qquad U\bar{U}^* = 0; \qquad U^*U + \bar{U}^*\bar{U} = NI \tag{4.9.87}$$

Hence,

$$\Pi_U^{\perp} = I - \frac{1}{N}U^*U = \frac{1}{N}\bar{U}^*\bar{U} \tag{4.9.88}$$

Also, note that (for $p = 1, \ldots, m$)

$$\begin{aligned}
w_k^p Y_k &= \sum_{t=0}^{N-1} y(t) e^{-i\frac{2\pi}{N}k(t-p)} \\
&= \sum_{t=0}^{p-1} y(t) w_k^{p-t} + \sum_{t=p}^{N-1} y(t) w_k^{N+p-t} \\
&= \big[y(p-1), \ldots, y(0), 0, \ldots, 0\big] \begin{bmatrix} w_k \\ \vdots \\ w_k^m \end{bmatrix} + \big[0, \ldots, 0, y(N-1), \ldots, y(p)\big] \begin{bmatrix} w_k \\ \vdots \\ w_k^N \end{bmatrix} \\
&\triangleq \mu_p^* u_k + \psi_p^* b_k
\end{aligned} \tag{4.9.89}$$

where u_k and b_k are as defined before (see (4.9.47) and (4.9.86)). Consequently, for $M = N$, the "data" matrix $Y\Pi_U^{\perp}$ used in the FRES-ESPRIT method can be written as (*cf.* (4.9.86)–(4.9.89))

$$\begin{aligned}
[u_1 Y_1, \ldots, u_N Y_N]\,\Pi_U^{\perp} &= \left\{ \begin{bmatrix} \mu_1^* \\ \vdots \\ \mu_m^* \end{bmatrix} [u_1, \ldots, u_N] + \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_m^* \end{bmatrix} [b_1, \ldots, b_N] \right\} \bar{U}^*\bar{U} \cdot \frac{1}{N} \\
&= \left\{ \begin{bmatrix} \mu_1^* \\ \vdots \\ \mu_m^* \end{bmatrix} U + \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_m^* \end{bmatrix} \begin{bmatrix} U \\ \bar{U} \end{bmatrix} \right\} \bar{U}^*\bar{U} \cdot \frac{1}{N} \\
&= \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_m^* \end{bmatrix} \begin{bmatrix} 0 \\ \bar{U} \end{bmatrix} = \begin{bmatrix} y(N-m) & \cdots & y(1) \\ y(N-m+1) & \cdots & y(2) \\ \vdots & & \vdots \\ y(N-1) & \cdots & y(m) \end{bmatrix} \bar{U}
\end{aligned} \tag{4.9.90}$$

It follows from (4.9.90) that the n principal (or dominant) left singular vectors of $Y\Pi_U^{\perp}$ are equal to the n principal eigenvectors of the following matrix (obtained by postmultiplying the right-hand side of (4.9.90) with its conjugate transpose and using the fact that $\bar{U}\bar{U}^* = NI$ from (4.9.87)):

$$\begin{bmatrix} y(N-m) & \cdots & y(1) \\ \vdots & & \vdots \\ y(N-1) & \cdots & y(m) \end{bmatrix} \begin{bmatrix} y^*(N-m) & \cdots & y^*(N-1) \\ \vdots & & \vdots \\ y^*(1) & \cdots & y^*(m) \end{bmatrix}$$

$$= \sum_{t=1}^{N-m} \begin{bmatrix} y(t) \\ \vdots \\ y(t+m-1) \end{bmatrix} \left[y^*(t), \ldots, y^*(t+m-1) \right] \tag{4.9.91}$$

which is precisely the type of sample covariance matrix used in the standard ESPRIT method. (Compare with (4.5.14); the difference between (4.9.91) and (4.5.14) is due to some notational changes made in this complement, such as in the definition of the matrix A.)

4.9.7 A Useful Result for Two-Dimensional (2D) Sinusoidal Signals

For a noise-free 1D sinusoidal signal

$$y(t) = \sum_{k=1}^{n} \beta_k e^{i\omega_k t}, \qquad t = 0, 1, 2, \ldots \tag{4.9.92}$$

a data vector of length m can be written as

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(m-1) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ e^{i\omega_1} & \cdots & e^{i\omega_n} \\ \vdots & & \vdots \\ e^{i(m-1)\omega_1} & \cdots & e^{i(m-1)\omega_n} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \triangleq A\beta \tag{4.9.93}$$

The matrix A just introduced is the complex conjugate of the one in (4.2.4). In this complement, we prefer to work with the type of A matrix in (4.9.93), to simplify the notation, but note that the discussion which follows applies without change to the complex conjugate of the above A as well (or, to its extension to 2D sinusoidal signals).

Let $\{c_k\}_{k=1}^n$ be defined uniquely via the equation

$$1 + c_1 z + \cdots + c_n z^n = \prod_{k=1}^{n} \left(1 - z e^{-i\omega_k}\right) \tag{4.9.94}$$

Then, it can be readily checked (see (4.5.21)) that the matrix

$$C^* = \begin{bmatrix} 1 & c_1 & \cdots & c_n & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & c_1 & \cdots & c_n \end{bmatrix}, \qquad (m-n) \times m \tag{4.9.95}$$

satisfies

$$C^*A = 0 \tag{4.9.96}$$

(To verify (4.9.96), it is enough to observe, from (4.9.94), that $1 + c_1 e^{i\omega_k} + \cdots + c_n e^{in\omega_k} = 0$ for $k = 1, \ldots, n$.) Furthermore, as $\operatorname{rank}(C) = m - n$ and $\dim[\mathcal{N}(A^*)] = m - n$ too, it follows from (4.9.96) that

$$\boxed{C \text{ is a basis for the null space of } A^*, \mathcal{N}(A^*)} \tag{4.9.97}$$

The matrix C plays an important role in the derivation and analysis of several frequency estimators. (See, e.g., Section 4.5, [Bresler and Macovski 1986], and [Stoica and Sharman 1990].)

In this complement, *we will extend the result* (4.9.97) *to 2D sinusoidal signals*. The derivation of a result similar to (4.9.97) for such signals is a rather more difficult problem than in the 1D case. The solution that we will present was introduced in [Clark and Scharf 1994]. (See also [Clark, Eldén, and Stoica 1997].) Using the extended result, we can derive parameter estimation methods for 2D sinusoidal signals in much the same manner as for 1D signals. (See the cited papers and Section 4.5.)

A noise-free 2D sinusoidal signal is described by the following equation (compare with (4.9.92)):

$$y(t, \bar{t}) = \sum_{k=1}^{n} \beta_k e^{i\omega_k t} e^{i\bar{\omega}_k \bar{t}}, \qquad t, \bar{t} = 0, 1, 2, \ldots \tag{4.9.98}$$

Let

$$\gamma_k = e^{i\omega_k}, \quad \lambda_k = e^{i\bar{\omega}_k} \tag{4.9.99}$$

Using this notation allows us to write (4.9.98) in the more compact form

$$y(t, \bar{t}) = \sum_{k=1}^{n} \beta_k \gamma_k^t \lambda_k^{\bar{t}} \tag{4.9.100}$$

Moreover, equation (4.9.100) (unlike (4.9.98)) also covers the case of *damped (2D) sinusoidal signals*, for which

$$\gamma_k = e^{\mu_k + i\omega_k}, \quad \lambda_k = e^{\bar{\mu}_k + i\bar{\omega}_k} \tag{4.9.101}$$

with $\{\mu_k, \bar{\mu}_k\}$ being the damping parameters ($\mu_k, \bar{\mu}_k \leq 0$).

The following notation will be used frequently in this complement:

$$g_t^* = \begin{bmatrix} \gamma_1^t & \cdots & \gamma_n^t \end{bmatrix} \tag{4.9.102}$$

$$\Gamma = \begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_n \end{bmatrix} \tag{4.9.103}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \tag{4.9.104}$$

$$\beta = \begin{bmatrix} \beta_1 & \dots & \beta_n \end{bmatrix}^T \tag{4.9.105}$$

$$A_L = \begin{bmatrix} 1 & \dots & 1 \\ \lambda_1 & \dots & \lambda_n \\ \vdots & & \vdots \\ \lambda_1^{L-1} & \dots & \lambda_n^{L-1} \end{bmatrix} \quad \text{for } L \geq n \tag{4.9.106}$$

Using (4.9.102), (4.9.104), and (4.9.105), we can write

$$y(t, \bar{t}) = g_t^* \Lambda^{\bar{t}} \beta \tag{4.9.107}$$

Hence, similarly to (4.9.93), we can write the $m\bar{m} \times 1$ data vector obtained from (4.9.98) for $t = 0, \dots, m-1$ and $\bar{t} = 0, \dots, \bar{m}-1$ as

$$\begin{bmatrix} y(0,0) \\ \vdots \\ y(0, \bar{m}-1) \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ y(m-1, 0) \\ \vdots \\ y(m-1, \bar{m}-1) \end{bmatrix} = \begin{bmatrix} g_0^* \Lambda^0 \\ \vdots \\ g_0^* \Lambda^{\bar{m}-1} \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ g_{m-1}^* \Lambda^0 \\ \vdots \\ g_{m-1}^* \Lambda^{\bar{m}-1} \end{bmatrix} \beta \triangleq \mathcal{A} \beta \tag{4.9.108}$$

The matrix $\mathcal{A}$ just defined,

$$\mathcal{A} = \begin{bmatrix} g_0^* \Lambda^0 \\ \vdots \\ g_0^* \Lambda^{\bar{m}-1} \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ g_{m-1}^* \Lambda^0 \\ \vdots \\ g_{m-1}^* \Lambda^{\bar{m}-1} \end{bmatrix} \quad (m\bar{m} \times n) \tag{4.9.109}$$

plays the same role for 2D sinusoidal signals as the matrix A in (4.9.93) does for 1D signals. Therefore, it is the null space of (4.9.109) that we want to characterize. More precisely, we want to find a *linearly parameterized basis* for the null space of the matrix $\mathcal{A}^*$ in (4.9.109), similar to the basis C for A^* in (4.9.93). (See (4.9.97).)

Note that, using (4.9.103), we can also write $y(t,\bar{t})$ as

$$y(t,\bar{t}) = \begin{bmatrix} \lambda_1^{\bar{t}} & \dots & \lambda_n^{\bar{t}} \end{bmatrix} \Gamma^t \beta \tag{4.9.110}$$

This means that $\mathcal{A}$ can also be written as follows:

$$\mathcal{A} = \begin{bmatrix} A_{\bar{m}}\Gamma^0 \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ A_{\bar{m}}\Gamma^{m-1} \end{bmatrix} \tag{4.9.111}$$

Similarly to (4.9.94), let us define the parameters $\{c_k\}_{k=1}^n$ uniquely via the equation

$$\boxed{1 + c_1 z + \cdots + c_n z^n = \prod_{k=1}^{n} \left(1 - \frac{z}{\lambda_k}\right)} \tag{4.9.112}$$

Note that there is a *one-to-one mapping between* $\{c_k\}$ *and* $\{\lambda_k\}$ $(\lambda_k \neq 0)$. In particular, we can obtain $\{\lambda_k\}$ uniquely from $\{c_k\}$. (See [Stoica and Sharman 1990] for more details on this aspect in the case of $\{\lambda_k = e^{i\omega_k}\}$.) Consequently, we can see the introduction of $\{c_k\}$ as a new parameterization of the problem, which replaces the parameterization via $\{\lambda_k\}$. Using $\{c_k\}$, we build the following matrix, similarly to (4.9.95), assuming $\bar{m} > n$:

$$C^* = \begin{bmatrix} 1 & c_1 & \cdots & c_n & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & c_1 & \cdots & c_n \end{bmatrix}, \qquad (\bar{m}-n) \times \bar{m} \tag{4.9.113}$$

We note (*cf.* (4.9.96)) that

$$C^* A_{\bar{m}} = 0 \tag{4.9.114}$$

It follows from (4.9.111) and (4.9.114) that

$$\underbrace{\begin{bmatrix} C^* & & 0 \\ & \ddots & \\ 0 & & C^* \end{bmatrix}}_{[m(\bar{m}-n)]\times m\bar{m}} \mathcal{A} = 0 \tag{4.9.115}$$

Hence, we have found $(m\bar{m} - mn)$ vectors of the sought basis for $\mathcal{N}(\mathcal{A}^*)$. It remains to find $(m-1)n$ additional (linearly independent) vectors of this basis (note that $\dim[\mathcal{N}(\mathcal{A}^*)] = m\bar{m} - n$). To find the remaining vectors, we need an approach that is rather different from that used so far.

Let us assume that

$$\boxed{\lambda_k \neq \lambda_p \text{ for } k \neq p} \tag{4.9.116}$$

and let the vector

$$b^* = [b_1, \ldots, b_n]$$

be defined via the linear (interpolation) equation

$$\boxed{b^* A_n = \left[\gamma_1, \ldots, \gamma_n\right]} \tag{4.9.117}$$

(with A_n as defined in (4.9.106)). *Under* (4.9.116) *and for given* $\{\lambda_k\}$*, there exists a one-to-one map between* $\{b_k\}$ *and* $\{\gamma_k\}$; hence, we can view the use of $\{b_k\}$ as a reparameterization of the problem. (Note that, if (4.9.116) does not hold, i.e., $\lambda_k = \lambda_p$, then, for identifiability reasons, we must have $\gamma_k \neq \gamma_p$, and therefore no vector b that satisfies (4.9.117) can exist.) From (4.9.117), we easily obtain

$$b^* A_n \Gamma^t = \left[\gamma_1, \ldots, \gamma_n\right] \Gamma^t = g_{t+1}^*$$

and hence (see also (4.9.109) and (4.9.111))

$$b^* \begin{bmatrix} g_t^* \Lambda^0 \\ \vdots \\ g_t^* \Lambda^{n-1} \end{bmatrix} = b^* A_n \Gamma^t = g_{t+1}^* \Lambda^0 \tag{4.9.118}$$

Next, we assume that

$$\bar{m} \geq 2n - 1 \tag{4.9.119}$$

which is a weak condition (typically we have $m, \bar{m} \gg n$). Under (4.9.119), we can write (making use of (4.9.118)):

$$\underbrace{\begin{bmatrix} b^* & & 0 \\ & \ddots & \\ 0 & & b^* \end{bmatrix}}_{B^*} \begin{bmatrix} g_t^* \Lambda^0 \\ \vdots \\ g_t^* \Lambda^{\bar{m}-1} \end{bmatrix} - \begin{bmatrix} g_{t+1}^* \Lambda^0 \\ \vdots \\ g_{t+1}^* \Lambda^{n-1} \end{bmatrix} = 0 \tag{4.9.120}$$

where

$$B^* = \begin{bmatrix} b_1 & b_2 & \dots & b_n & 0 & \dots & 0 \\ & \ddots & & & \ddots & \ddots & \vdots \\ 0 & & b_1 & b_2 & \dots & b_n & 0 \end{bmatrix} \qquad (n \times \bar{m})$$

Note that, indeed, we need $\bar{m} \geq 2n - 1$ to be able to write (4.9.120) (if $\bar{m} > 2n - 1$, then the rightmost $\bar{m} - 2n - 1$ columns of B^* are zeroes). Combining (4.9.115) and (4.9.120) yields the following matrix, whose rows lie in the left null space of $\mathcal{A}$:

$$\boxed{\left.\begin{bmatrix} \mathcal{D} & \mathcal{I} & & & \\ & \mathcal{D} & \mathcal{I} & & 0 \\ & & \ddots & \ddots & \\ 0 & & & \mathcal{D} & \mathcal{I} \\ & & & & C^* \end{bmatrix}\right\} \begin{matrix} m \\ \text{block rows} \end{matrix}} \tag{4.9.121}$$

where

$$\mathcal{D} = \begin{bmatrix} C^* \\ B^* \end{bmatrix} = \left[\begin{array}{cccccc} 1 & c_1 & \cdots & c_n & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & c_1 & \cdots & c_n \\ \hline b_1 & \dots & b_n & 0 & \dots & 0 \\ & \ddots & & \ddots & \ddots & \vdots \\ 0 & & b_1 & \dots & b_n & 0 \end{array}\right] \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} \bar{m} - n \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} n \end{matrix} \qquad (\bar{m} \times \bar{m})$$

$$\mathcal{I} = \left[\begin{array}{ccccc} 0 & & \cdots & & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & 0 \\ \hline -1 & 0 & & \dots & 0 \\ & \ddots & \ddots & & \vdots \\ 0 & & -1 & 0 \; \dots & 0 \end{array}\right] \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} \bar{m} - n \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} n \end{matrix} \qquad (\bar{m} \times \bar{m})$$

The matrix in (4.9.121) is of dimension $[(m-1)\bar{m} + (\bar{m} - n)] \times m\bar{m}$, that is $(m\bar{m} - n) \times m\bar{m}$, and its rank is equal to $m\bar{m} - n$ (i.e., it has full row rank, as $c_n \neq 0$). Consequently, *the rows of* (4.9.121) *form a linearly parameterized basis for the null space of* $\mathcal{A}$. We remind the reader that, under (4.9.116), there is a one-to-one map between $\{\lambda_k, \gamma_k\}$ and the basis parameters $\{c_k, b_k\}$. (See (4.9.112) and (4.9.117).) Hence, we can think of estimating $\{c_k, b_k\}$ in lieu of $\{\lambda_k, \gamma_k\}$, at least in a first stage, and, when we do so, the linear dependence of (4.9.121) on the unknown parameters will come in quite handy. As a simple example of such an estimation method based on (4.9.121), note that the modified MUSIC procedure outlined in Section 4.5 can easily be extended to the case of 2D signals by making use of (4.9.121).

Compared with the basis matrix for the 1D case (see (4.9.95)), the null space basis (4.9.121) in the 2D case is apparently much more complicated. In addition, the given 2D basis result depends on the condition (4.9.116); if (4.9.116) is even approximately violated (i.e., if there exist λ_k and λ_p with $k \neq p$ such that $\lambda_k \simeq \lambda_p$), then the mapping $\{\gamma_k\} \leftrightarrow \{b_k\}$ could become ill-conditioned and so cause a deterioration of the estimation accuracy.

Finally, we remark on the fact that, for damped sinusoids, the parameterization via $\{b_k\}$ and $\{c_k\}$ is parsimonious. However, for undamped sinusoidal signals, the parameterization via $\{\omega_k, \bar{\omega}_k\}$ contains $2n$ real-valued unknowns, whereas the one based on $\{b_k, c_k\}$ has $4n$ unknowns, or $3n$ unknowns if a certain conjugate symmetry property of $\{b_k\}$ is exploited (see, e.g., [Stoica and Sharman 1990]); hence, in such a case the use of $\{b_k\}$ and, in particular, $\{c_k\}$ leads to an overparameterized problem, which might also result in a (slight) accuracy degradation. The previous criticism of the result (4.9.121) is, however, minor, and, in fact, (4.9.121) is the *only* known basis for $\mathcal{N}(\mathcal{A}^*)$.

4.10 EXERCISES

Exercise 4.1: Speed Measurement by a Doppler Radar as a Frequency Estimation Problem
Assume that a radar system transmits a sinusoidal signal towards an object. For the sake of simplicity, further assume that the object moves along a trajectory parallel to the wave propagation direction, at a constant velocity v. Let $\alpha e^{i\omega t}$ denote the signal emitted by the radar. Show that the backscattered signal, measured by the radar system after reflection off the object, is given by

$$s(t) = \beta e^{i(\omega - \omega^D)t} + e(t) \tag{4.10.1}$$

where $e(t)$ is measurement noise, ω^D is the so-called *Doppler frequency*,

$$\omega^D \triangleq 2\omega v/c$$

and

$$\beta = \mu\alpha e^{-2i\omega r/c}$$

Here c denotes the speed of wave propagation, r is the object range, and μ is an attenuation coefficient.

Conclude from (4.10.1) that the problem of speed measurement can be reduced to one of frequency determination. The latter problem can be solved by using the methods of this chapter.

Exercise 4.2: ACS of Sinusoids with Random Amplitudes or Nonuniform Phases
In some applications, it is not reasonable to assume that the amplitudes of the sinusoidal terms are fixed or that their phases are uniformly distributed. Examples are fast fading in mobile telecommunications (where the amplitudes vary), and sinusoids that have been tracked so that their phase is random, near zero, but not uniformly distributed. We derive the ACS for such cases.

Let $x(t) = \alpha e^{i(\omega_0 t + \varphi)}$, where α and φ are statistically independent random variables and ω_0 is a constant. Assume that α has mean $\bar{\alpha}$ and variance σ_α^2.

(a) If φ is uniformly distributed on $[-\pi, \pi]$, find $E\{x(t)\}$ and $r_x(k)$. Show also that, if α is constant, the expression for $r_x(k)$ reduces to equation (4.1.5).

(b) If φ is not uniformly distributed on $[-\pi, \pi]$, express $E\{x(t)\}$ in terms of the probability density function $p(\varphi)$. Find sufficient conditions on $p(\varphi)$ such that $x(t)$ is zero mean, find $r_x(k)$ in this case, and give an example of such a $p(\varphi)$.

Exercise 4.3: A Nonergodic Sinusoidal Signal

As shown in Complement 4.9.1, the signal

$$x(t) = \alpha e^{i(\omega t + \varphi)}$$

with α and ω being nonrandom constants and φ being uniformly distributed on $[0,\ 2\pi]$, is second-order ergodic in the sense that the mean and covariances determined from an (infinitely long) temporal realization of the signal coincide with the mean and covariances obtained from an ensemble of (infinitely many) realizations. In the present exercise, assume that α and ω are independent random variables, with ω being uniformly distributed on $[0,\ 2\pi]$; the initial-phase variable φ may be arbitrarily distributed (in particular it can be nonrandom). Show that, in such a case,

$$E\left\{x(t)x^*(t-k)\right\} = \begin{cases} E\left\{\alpha^2\right\} & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \tag{4.10.2}$$

Also, show that the covariances obtained by "temporal averaging" differ from those given, and hence deduce that the signal is not ergodic. Comment on the behavior of such a signal over the ensemble of realizations and in each realization, respectively.

Exercise 4.4: AR Model-Based Frequency Estimation

Consider the noisy sinusoidal signal

$$y(t) = x(t) + e(t)$$

where $x(t) = \alpha e^{i(\omega_0 t + \varphi)}$ (with $\alpha > 0$ and φ uniformly distributed on $[0,\ 2\pi]$) and $e(t)$ is white noise with zero mean and unit variance. An AR model of order $n \geq 1$ is fitted to $\{y(t)\}$ by using the Yule–Walker or LS method. In the limiting case of an infinitely long data sample, the AR coefficients are given by the solution to (3.4.4). Show that the PSD, corresponding to the AR model determined from (3.4.4), has a global peak at $\omega = \omega_0$. Conclude that AR modeling can be used in this case to find the sinusoidal frequency, in spite of the fact that $\{y(t)\}$ does not satisfy an AR equation of finite order. (In the case of multiple sinusoids, the AR frequency estimates are biased.) Regarding the estimation of the signal power, however, show that the height of the global peak of the AR spectrum does not directly provide an "estimate" of α^2.

Exercise 4.5: An ARMA Model-Based Derivation of the Pisarenko Method

Let R denote the covariance matrix (4.2.7) with $m = n + 1$, and let g be the eigenvector of R associated with its minimum eigenvalue. The Pisarenko method determines the signal frequencies by exploiting the fact that

$$a^*(\omega)g = 0 \qquad \text{for } \omega = \omega_k,\ k = 1, \ldots, n \tag{4.10.3}$$

(*cf.* (4.5.13) and (4.5.17)). Derive the property (4.10.3) directly from the ARMA model equation (4.2.3).

Exercise 4.6: Frequency Estimation when Some Frequencies Are Known
Assume that $y(t)$ is known to have p sinusoidal components at known frequencies $\{\tilde{\omega}_k\}_{k=1}^{p}$ (but with unknown amplitudes and phases), plus $n - p$ other sinusoidal components whose frequencies are unknown. Develop a modification of the HOYW method to estimate the unknown frequencies from measurements $\{y(t)\}_{t=1}^{N}$ without estimating the known frequencies.

Exercise 4.7: A Combined HOYW–ESPRIT Method for the MA Noise Case
The HOYW method, presented in Section 4.4 for the white-noise case, is based on the matrix Γ in (4.2.8). Let us assume that the noise sequence $\{e(t)\}$ in (4.1.1) is known to be an MA process of order m and that m is given. A simple way to handle such a colored noise in the HOYW method consists of modifying the expression (4.2.8) of Γ as follows:

$$\tilde{\Gamma} = E\left\{ \begin{bmatrix} y(t-L-1-m) \\ \vdots \\ y(t-L-M-m) \end{bmatrix} [y^*(t), \ldots, y^*(t-L)] \right\} \tag{4.10.4}$$

Derive an expression for $\tilde{\Gamma}$ similar to the one for Γ in (4.2.8). Furthermore, make use of that expression in an ESPRIT-like method to estimate the frequencies $\{\omega_k\}$, instead of using it in an HOYW-like method (as in Section 4.4). Discuss the advantage of this so-called HOYW–ESPRIT method over the HOYW method based on $\tilde{\Gamma}$. Assuming that the noise is white (i.e., $m = 0$) and hence that ESPRIT is directly applicable, would you prefer using HOYW–ESPRIT (with $m = 0$) in lieu of ESPRIT? Why or why not?

Exercise 4.8: Chebyshev Inequality and the Convergence of Sample Covariances
Let x be a random variable with finite mean μ and variance σ^2. Show that, for any positive constant c, the so-called Chebyshev inequality holds:

$$\boxed{\Pr(|x-\mu| \geq c\sigma) \leq 1/c^2} \tag{4.10.5}$$

Use (4.10.5) to show that, if a sample covariance lag $\hat{r}_N$ (estimated from N data samples) converges to the true value r *in the mean-square sense*

$$\lim_{N\to\infty} E\left\{|\hat{r}_N - r|^2\right\} = 0 \tag{4.10.6}$$

then $\hat{r}_N$ also converges to r *in probability*:

$$\lim_{N\to\infty} \Pr(|\hat{r}_N - r| \neq 0) = 0 \tag{4.10.7}$$

For sinusoidal signals, the mean-square convergence of $\{\hat{r}_N(k)\}$ to $\{r(k)\}$, as $N \to \infty$, has been proven in Complement 4.9.1. (In this exercise, we omit the argument k in $\hat{r}_N(k)$ and $r(k)$, for notational simplicity.) Additionally, discuss the use of (4.10.5) to set *bounds* (which hold with a specified probability) on an arbitrary random variable with given mean and variance. Comment on the conservatism of the bounds obtained from (4.10.5) by comparing them with the bounds corresponding to a Gaussian random variable.

Exercise 4.9: More about the Forward–Backward Approach

The sample covariance matrix in (4.8.3), used by the forward–backward approach, is often a better estimate of the theoretical covariance matrix than $\hat{R}$ is (as argued in Section 4.8). Another advantage of (4.8.3) is that the forward–backward sample covariance is always numerically better conditioned than the usual (forward-only) sample covariance matrix $\hat{R}$. To understand this statement, let R be a Hermitian matrix (not necessarily a Toeplitz one, like the R in (4.2.7)). The "condition number" of R is defined as

$$\text{cond}(R) = \lambda_{\max}(R)/\lambda_{\min}(R)$$

where $\lambda_{\max}(R)$ and $\lambda_{\min}(R)$ are the maximum and minimum eigenvalues of R, respectively. The numerical errors that affect many algebraic operations on R, such as inversion, eigendecomposition, and so on, are essentially proportional to cond(R). Hence, the smaller cond(R), the better. (See Appendix A for details on this aspect.)

Next, let U be a unitary matrix (the J in (4.8.3) being a special case of such a matrix). Observe that the forward–backward covariance in equation (4.8.3) is of the form $R + U^* R^T U$. Prove that

$$\boxed{\text{cond}(R) \geq \text{cond}(R + U^* R^T U)} \tag{4.10.8}$$

for any unitary matrix U. We note that the result (4.10.8) applies to any Hermitian matrix R and unitary matrix U, and thus is valid in cases more general than the forward–backward approach in Section 4.8, in which R is Toeplitz and $U = J$.

Exercise 4.10: ESPRIT and Min–Norm Under the Same Umbrella

ESPRIT and Min–Norm methods are seemingly quite different from one another; it might well seem unlikely that there is any strong relationship between them. It is the goal of this exercise to show that in fact ESPRIT and Min–Norm are quite related closely to each other. We will see that ESPRIT and Min–Norm are members of a well-defined class of frequency estimates.

Consider the equation

$$\hat{S}_2^* \hat{\Psi} = \hat{S}_1^* \tag{4.10.9}$$

where $\hat{S}_1$ and $\hat{S}_2$ are as defined in Section 4.7. The $(m-1) \times (m-1)$ matrix $\hat{\Psi}$ in (4.10.9) is the unknown. First, show that the asymptotic counterpart of (4.10.9),

$$S_2^* \Psi = S_1^* \tag{4.10.10}$$

has the property that any of its solutions Ψ has n eigenvalues equal to $\{e^{-i\omega_k}\}_{k=1}^n$. This property, along with the fact that there is an infinite number of matrices $\hat{\Psi}$ satisfying (4.10.9) (see

Section A.8 in Appendix A), implies that (4.10.9) generates a class of frequency estimators with an infinite number of members.

As a second task, show that ESPRIT and Min–Norm belong to this class of estimators. In other words, prove that there is a solution of (4.10.9) whose nonzero eigenvalues have exactly the same arguments as the eigenvalues of the ESPRIT matrix $\hat{\phi}$ in (4.7.12), and also that there is another solution of (4.10.9) whose eigenvalues are equal to the roots of the Min–Norm polynomial in (4.6.3). For more details on the topic of this exercise, see [Hua and Sarkar 1990].

Exercise 4.11: Yet Another Relationship between ESPRIT and Min–Norm

Let the vector $[\hat{\rho}^T, 1]^T$ be defined similarly to the Min–Norm vector $[1, \hat{g}^T]^T$ (see (4.6.1)), the only difference being that now we constrain the last element to be equal to one. Hence, $\hat{\rho}$ is the minimum-norm solution to (see (4.6.5))

$$\hat{S}^* \begin{bmatrix} \hat{\rho} \\ 1 \end{bmatrix} = 0$$

Use the Min–Norm vector $\hat{\rho}$ to build the following matrix

$$\tilde{\phi} = \hat{S}^* \left[\begin{array}{c|c} 0 & \begin{array}{c} I_{m-1} \\ \hline -\hat{\rho}^* \end{array} \end{array} \right] \hat{S} \quad (n \times n)$$

Prove the somewhat curious fact that $\tilde{\phi}$ above is equal to the ESPRIT matrix, $\hat{\phi}$, in (4.7.12).

COMPUTER EXERCISES

Tools for Frequency Estimation:

The text website www.prenhall.com/stoica contains the following MATLAB functions for use in computing frequency estimates and estimating the number of sinusoidal terms. In the first four functions, y is the data vector and n is the desired number of frequency estimates. The remaining variables are described below.

- `w=hoyw(y,n,L,M)`
  The HOYW estimator given in the box on page 166; L and M are the matrix dimensions as in (4.4.8).
- `w=music(y,n,m)`
  The Root MUSIC estimator given by (4.5.12); m is the dimension of $a(\omega)$. This function also implements the Pisarenko method by setting $m = n + 1$.
- `w=minnorm(y,n,m)`
  The Root Min–Norm estimator given by (4.6.3); m is the dimension of $a(\omega)$.
- `w=esprit(y,n,m)`
  The ESPRIT estimator given by (4.7.12); m is the size of the square matrix $\hat{R}$ there, and S_1 and S_2 are chosen as in equations (4.7.5) and (4.7.6).

- `order=sinorder(mvec,sig2,N,nu)`
  Computes the AIC, AIC$_c$, GIC, and BIC model-order selections for sinusoidal parameter estimation problems. See Appendix C for details on the derivations of these methods. Here, `mvec` is a vector of candidate sinusoidal model orders, `sig2` is the vector of estimated residual variances corresponding to the model orders in `mvec`, `N` is the length of the observed data vector, and `nu` is a parameter in the GIC method. The 4-element output vector `order` contains the selected model orders obtained from AIC, AIC$_c$, GIC, and BIC, respectively.

Exercise C4.12: Resolution Properties of Subspace Methods for Estimation of Line Spectra
In this exercise, we test and compare the resolution properties of four subspace methods: Min–Norm, MUSIC, ESPRIT, and HOYW.

Generate realizations of the sinusoidal 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 $N = 64$, $e(t)$ is Gaussian white noise with variance σ^2, and $\varphi_1,\ \varphi_2$ are independent random variables each uniformly distributed on $[-\pi, \pi]$.

Generate 50 Monte Carlo realizations of $y(t)$, and present the results from these experiments. The results of frequency estimation can be presented, comparing the sample means and variances of the frequency estimates from the various estimators.

(a) Find the exact ACS for $y(t)$. Compute the "true" frequency estimates from the four methods, for $n = 4$ and various choices of the order $m \geq 5$ (and corresponding choices of M and L for HOYW). Which method(s) are able to resolve the two sinusoids, and for what values of m (or M and L)?

(b) Consider now $N = 64$, and set $\sigma^2 = 0$; this corresponds to the case of finite data length but infinite SNR. Compute frequency estimates for the four techniques again, using $n = 4$ and various choices of m, M, and L. Which method(s) are reliably able to resolve the sinusoids? Explain why.

(c) Obtain frequency estimates from the four methods when $N = 64$ and $\sigma^2 = 1$. Use $n = 4$, and experiment with different choices of m, M, and L to see the effect on estimation accuracy (e.g., try $m = 5$, 8, and 12 for MUSIC, Min–Norm, and ESPRIT, and try $L = M = 4$, 8, and 12 for HOYW). Which method(s) give reliable "superresolution" estimation of the sinusoids? Is it possible to resolve the two sinusoids in the signal? Discuss how the choices of m, M, and L influence the resolution properties. Which method appears to have the best resolution?

You may want to experiment further by changing the SNR and the relative amplitudes of the sinusoids to gain a better understanding of the differences between the methods.

(d) Compare the estimation results with the AR and ARMA results obtained in Exercise C3.18 in Chapter 3. What are the major differences between the techniques? Which method(s) do you prefer for this problem?

Exercise C4.13: Model Order Selection for Sinusoidal Signals

In this exercise, we examine four methods for model order selection for sinusoidal signals. As discussed in Appendix C, several important model order selection rules have the general form (see (C.8.1)–(C.8.2))

$$-2 \ln p_n(y, \hat{\theta}^n) + \eta(r, N) r \tag{4.10.11}$$

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

$$\begin{aligned} \text{AIC}: \quad & \eta(r, N) = 2 \\ \text{AIC}_\text{c}: \quad & \eta(r, N) = 2\frac{N}{N - r - 1} \\ \text{GIC}: \quad & \eta(r, N) = \nu \ (e.g., \nu = 4) \\ \text{BIC}: \quad & \eta(r, N) = \ln N \end{aligned} \tag{4.10.12}$$

Here, N is the length of the observed data vector y and, for sinusoidal signals, r is given (see Appendix C) by

$$\begin{aligned} r &= 3n + 1 \quad \text{for AIC, AIC}_\text{c}\text{, and GIC} \\ r &= 5n + 1 \quad \text{for BIC} \end{aligned}$$

where n is the number of sinusoids in the model. 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 order of n; it is given (*cf.* (C.2.7)–(C.2.8) in Appendix C) by

$$-2 \ln p_n(y, \hat{\theta}^n) = N \hat{\sigma}_n^2 + \text{constant} \tag{4.10.13}$$

where

$$\hat{\sigma}_n^2 = \frac{1}{N} \sum_{t=1}^{N} \left| y(t) - \sum_{k=1}^{n} \hat{\alpha}_k e^{i(\hat{\omega}_k t + \hat{\varphi}_k)} \right|^2 \tag{4.10.14}$$

The selected model order is the value of n that minimizes (4.10.11). The preceding order selection rules, although derived for ML estimates of θ, can be used even with approximate ML estimates of θ, albeit with some loss of performance.

Well-Separated Sinusoids.

(a) Generate 100 realizations of

$$y(t) = 10 \sin[2\pi f_0 t + \varphi_1] + 5 \sin[2\pi (f_0 + \Delta f) t + \varphi_2] + e(t), \qquad t = 1, \ldots, N$$

for $f_0 = 0.24$, $\Delta f = 3/N$, and $N = 128$. Here, $e(t)$ is real-valued white noise with variance σ^2. For each realization, generate φ_1 and φ_2 as random variables uniformly distributed on $[0, 2\pi]$.

(b) Set $\sigma^2 = 10$. For each realization, estimate the frequencies of $n = 1, \ldots, 10$ real-valued sinusoidal components, by using ESPRIT, and estimate the amplitudes and phases by using the second equation in (4.3.8), where $\hat{\omega}$ is the vector of ESPRIT frequency estimates. Note that you will need to use two complex exponentials to model each real-valued sinusoid, so the number of frequencies to estimate with ESPRIT will be $2, 4, \ldots, 20$; however, the frequency estimates will be in symmetric pairs. Use $m = 40$ as the covariance matrix size in ESPRIT.

(c) Find the model orders that minimize AIC, AIC$_c$, GIC (with $\nu = 4$), and BIC. For each of the four order-selection methods, plot a histogram of the selected orders for the 100 realizations. Comment on their relative performance.

(d) Repeat the preceding experiment, using $\sigma^2 = 1$ and $\sigma^2 = 0.1$, and comment on the performance of the order selection methods as a function of SNR.

Closely Spaced Sinusoids. Generate 100 realizations of $y(t)$ as in previous case, but this time using $\Delta f = 0.5/N$. Repeat the preceding experiments. In addition, compare the relative performance of the order selection methods for well-separated versus closely spaced sinusoidal signals.

Exercise C4.14: Line Spectral Methods Applied to Measured Data

Apply the Min–Norm, MUSIC, ESPRIT, and HOYW frequency estimators to the data in the files `sunspotdata.mat` and `lynxdata.mat` (using both the original `lynx` data and the logarithmically transformed data, as in Exercise C2.23). These files can be obtained from the text website `www.prenhall.com/stoica`. Try to answer the following questions:

(a) Is the sinusoidal model appropriate for the data sets under study?

(b) Suggest how to choose the number of sinusoids in the model. (See Exercise C4.13.)

(c) What periodicities can you find in the two data sets?

Compare the results you obtain here to the AR(MA) and nonparametric spectral estimation results you obtained in Exercises C2.23 and C3.20.

5

Filter-Bank Methods

5.1 INTRODUCTION

The problem of estimating the PSD function $\phi(\omega)$ of a signal from a finite number of observations N is ill posed from a statistical standpoint, *unless* we make some appropriate assumptions on $\phi(\omega)$. More precisely, without any assumption on the PSD, we are required to estimate an *infinite* number of independent values $\{\phi(\omega)\}_{\omega=-\pi}^{\pi}$ from a *finite* number of samples. Evidently, we cannot do that in a consistent manner. In order to overcome this problem, we can either

$$\boxed{\text{parameterize } \{\phi(\omega)\} \text{ by means of a finite–dimensional model}} \tag{5.1.1}$$

or

$$\boxed{\begin{array}{l}\text{smooth the set } \{\phi(\omega)\}_{\omega=-\pi}^{\pi} \text{ by assuming that } \phi(\omega) \text{ is constant (or nearly} \\ \text{constant) over the band } [\omega - \beta\pi, \omega + \beta\pi], \text{ for some given } \beta \ll 1.\end{array}} \tag{5.1.2}$$

The approach based on (5.1.1) leads to the parametric spectral methods of Chapters 3 and 4, for which the estimation of $\{\phi(\omega)\}$ is reduced to the problem of estimating a number of parameters that is usually much smaller than the data length N.

The other approach to PSD estimation, (5.1.2), leads to the methods to be described in this chapter. The nonparametric methods of Chapter 2 are also (implicitly) based on (5.1.2), as is shown in Section 5.2. The approach (5.1.2) should, of course, be used for PSD estimation when we do not have enough information about the studied signal to be able to describe it (and its PSD) by a simple model (such as the ARMA equation in Chapter 3 or the equation of superimposed sinusoidal signals in Chapter 4). On the one hand, this implies that the methods derived from (5.1.2) can be used in cases where those based on (5.1.1) cannot.[1] On the other hand, we should expect to pay some price in using (5.1.2) over (5.1.1). Under the assumption in (5.1.2), $\phi(\omega)$ is described by $2\pi/2\pi\beta = 1/\beta$ values. In order to estimate these values from the available data in a consistent manner, we must require that $1/\beta < N$ or

$$\boxed{N\beta > 1} \tag{5.1.3}$$

As β increases, the achievable statistical accuracy of the estimates of $\{\phi(\omega)\}$ should increase (because the number of PSD values estimated from the given N data samples decreases) but the resolution decreases (because $\phi(\omega)$ is assumed to be constant on a larger interval). This *tradeoff between statistical variability and resolution* is the price paid for the generality of the methods derived from (5.1.2). We have already met this tradeoff in our discussion of the periodogram-based methods in Chapter 2. Note, from (5.1.3), that the *resolution threshold* β of the methods based on (5.1.2) can be lowered down to $1/N$ only if we are going to accept a significant statistical variability for our spectral estimates (because for $\beta = 1/N$ we will have to estimate N spectral values from the available N data samples). The parametric (or model-based) approach embodied in (5.1.1) describes the PSD by a number of parameters that is often much smaller than N; yet, it might achieve better resolution (i.e., a resolution threshold less than $1/N$) compared to the approach derived from (5.1.2).

When taking the approach (5.1.2) to PSD estimation, we are basically following the "definition" (1.1.1) of the spectral estimation problem, which we restate here (in abbreviated form) for easy reference:

$$\boxed{\text{From a finite-length data sequence, estimate how the power is distributed over narrow spectral bands.}} \tag{5.1.4}$$

There is an implicit assumption in (5.1.4) that the power is (nearly) constant over "narrow spectral bands," which is a restatement of (5.1.2).

The most natural implementation of the approach to spectral estimation resulting from (5.1.2) and (5.1.4) is depicted in Figure 5.1. The bandpass filter in this figure, which sweeps through the frequency interval of interest, can be viewed as a bank of (bandpass) filters. This observation motivates the name *filter-bank approach* given to the PSD estimation scheme sketched in Figure 5.1. Depending on the bandpass filter chosen, we obtain various filter-bank methods of spectral estimation. Even for a given bandpass filter, we may implement the scheme of Figure 5.1 in different ways, which leads to an even richer class of methods. Examples of bandpass filters that can be used in the scheme of Figure 5.1, as well as specific ways in which they may be

[1] This statement should be interpreted with some care. One can certainly use, for instance, an ARMA spectral model even if one does not know that the studied signal is really an ARMA signal. However, in such a case, one not only needs to estimate the model parameters, but also must face the rather difficult task of determining the structure of the parametric model used (for example, the orders of the ARMA model). The nonparametric approach to PSD estimation does not require any structure-determination step.

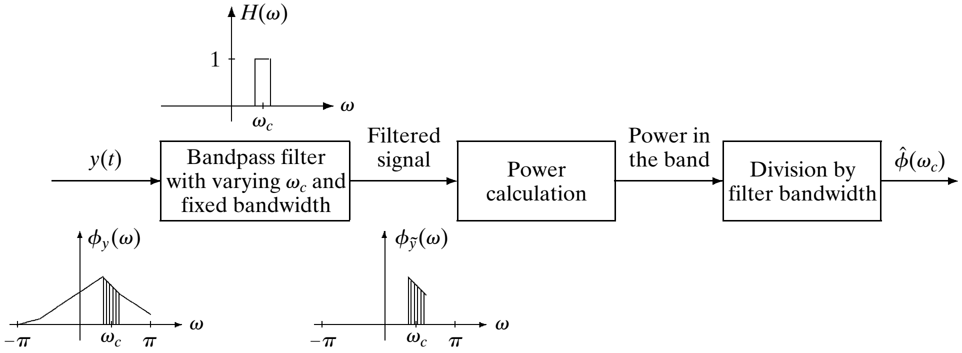


Figure 5.1 The filter-bank approach to PSD estimation.

implemented, are given in the remainder of this chapter. First, however, we discuss a few more aspects regarding the scheme in Figure 5.1.

As a mathematical motivation of the filter-bank approach (FBA) to spectral estimation, we prove the following result:

Assume that

(i) $\phi(\omega)$ is (nearly) constant over the filter passband;
(ii) the filter gain is (nearly) 1 over the passband and (nearly) zero outside the passband; *and*
(iii) the power of the filtered signal is a consistent estimate of the true power. (5.1.5)

Then

the PSD estimate, $\hat{\phi}_{\text{FB}}(\omega)$, obtained with the filter-bank approach, is a good approximation of $\phi(\omega)$.

Let $H(\omega)$ denote the transfer function of the bandpass filter, and let $2\pi\beta$ denote its bandwidth. Then, by using the formula (1.4.9) and the assumptions (iii), (ii), and (i) (in that order), we can write

$$\begin{aligned}
\hat{\phi}_{\text{FB}}(\omega) &\simeq \frac{1}{2\pi\beta}\int_{-\pi}^{\pi} |H(\psi)|^2 \phi(\psi)\, d\psi \\
&\simeq \frac{1}{2\pi\beta}\int_{\omega-\beta\pi}^{\omega+\beta\pi} \phi(\psi)\, d\psi \simeq \frac{1}{2\pi\beta} 2\pi\beta\phi(\omega) = \phi(\omega)
\end{aligned} \tag{5.1.6}$$

where ω denotes the center frequency of the bandpass filter. This is the result that we set out to prove.

If all three assumptions in (5.1.5) could be satisfied, then the FBA methods would produce spectral estimates with high resolution and low statistical variability. Unfortunately, these assumptions contain conflicting requirements that cannot be met simultaneously. In high-resolution applications, assumption (i) can be satisfied if we use a filter with a *very sharp passband*. According to the time-bandwidth product result (2.6.5), such a filter has a very long impulse response. This implies that we might not be able to get more than a few samples of the filtered signal (sometimes only one sample, see Section 5.2!). Hence, assumption (iii) cannot be met. In order to satisfy (iii), we need to average many samples of the filtered signal and, therefore, should consider a bandpass filter with a relatively short impulse response and, hence, a not too narrow passband. Assumption (i) may then be violated or, in other words, the resolution might be sacrificed.

The above discussion has brought once more to light *the compromise between resolution and statistical variability* and the fact that *the resolution is limited by the sample length*. These are the critical issues for any PSD estimation method based on the approach (5.1.2), such as those of Chapter 2 and the ones discussed in the following sections. These two issues will always surface within the nonparametric approach to spectral estimation—in many different ways, depending on the specific method at hand.

5.2 FILTER-BANK INTERPRETATION OF THE PERIODOGRAM

The value of the basic periodogram estimator (2.2.1) at a given frequency, say $\tilde{\omega}$, can be expressed as

$$\begin{aligned}\hat{\phi}_p(\tilde{\omega}) &= \frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\tilde{\omega}t}\right|^2 = \frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{i\tilde{\omega}(N-t)}\right|^2 \\ &= \frac{1}{\beta}\left|\sum_{k=0}^{N-1} h_k y(N-k)\right|^2 \end{aligned} \tag{5.2.1}$$

where $\beta = 1/N$ and

$$h_k = \frac{1}{N}e^{i\tilde{\omega}k} \qquad k = 0, \ldots, N-1 \tag{5.2.2}$$

The *truncated* convolution sum that appears in (5.2.1) can be written as the usual convolution sum associated with a linear causal system, if the weighting sequence in (5.2.2) is padded with zeroes:

$$y_F(N) = \sum_{k=0}^{\infty} h_k y(N-k) \tag{5.2.3}$$

with

$$h_k = \begin{cases} e^{i\tilde{\omega}k}/N & \text{for } k = 0, \ldots, N-1 \\ 0 & \text{otherwise} \end{cases} \tag{5.2.4}$$

The transfer function (or the frequency response) of the linear filter corresponding to $\{h_k\}$ in (5.2.4) is readily evaluated:

$$H(\omega) = \sum_{k=0}^{\infty} h_k e^{-i\omega k} = \frac{1}{N}\sum_{k=0}^{N-1} e^{i(\tilde{\omega}-\omega)k} = \frac{1}{N}\frac{e^{iN(\tilde{\omega}-\omega)}-1}{e^{i(\tilde{\omega}-\omega)}-1}$$

which gives

$$\boxed{H(\omega) = \frac{1}{N}\frac{\sin[N(\tilde{\omega}-\omega)/2]}{\sin[(\tilde{\omega}-\omega)/2]}e^{i(N-1)(\tilde{\omega}-\omega)/2}} \tag{5.2.5}$$

Figure 5.2 shows $|H(\omega)|$ as a function of $\Delta\omega = \tilde{\omega} - \omega$, for $N = 50$. It can be seen that $H(\omega)$ in (5.2.5) is the transfer function of a bandpass filter with center frequency equal to $\tilde{\omega}$. The *3-dB bandwidth* of this filter can be shown to be approximately $2\pi/N$ radians per sampling interval, or $1/N$ cycles per sampling interval. In fact, by comparing (5.2.5) to (2.4.17), we see that $H(\omega)$ resembles the DTFT of the rectangular window, the only differences being the phase term (due to the time offset) and the window lengths ($(2N - 1)$ in (2.4.17) versus N in (5.2.5)).

Thus, we have proven the following *filter-bank interpretation of the basic periodogram*:

> The periodogram $\hat{\phi}_p(\omega)$ can be exactly obtained by the FBA in Figure 5.1, where the bandpass filter's frequency response is given by (5.2.5), its bandwidth is $1/N$ cycles per sampling interval, and the power calculation is done from a *single sample* of the filtered signal. (5.2.6)

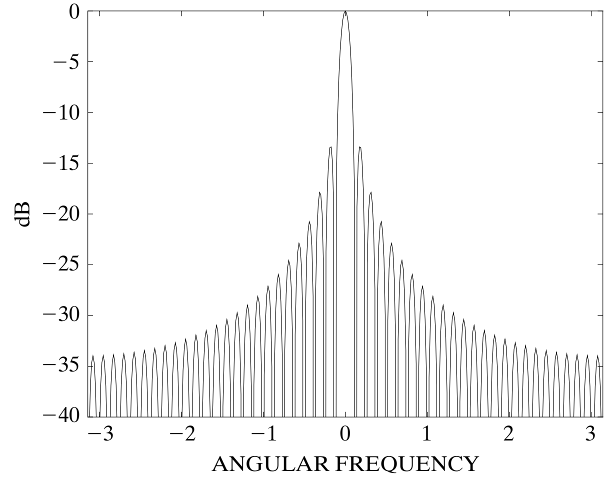


Figure 5.2 The magnitude of the frequency response of the bandpass filter $H(\omega)$ in (5.2.5), associated with the periodogram ($N = 50$), plotted as a function of $(\tilde{\omega} - \omega)$.

This interpretation of $\hat{\phi}_p(\omega)$ highlights a conclusion that is reached, in a different way, in Chapter 2: *the unmodified periodogram sacrifices statistical accuracy for resolution*. Indeed, $\hat{\phi}_p(\omega)$ uses a bandpass filter with the smallest bandwidth afforded by a time aperture of length N. In this way, it achieves a good resolution (see assumption (i) in (5.1.5)). The consequence of doing so is that only one (filtered) data sample is obtained for the power-calculation stage, which explains the erratic fluctuations of $\hat{\phi}_p(\omega)$ (owing to violation of assumption (iii) in (5.1.5)).

As explained in Chapter 2, the *modified periodogram methods* (Bartlett, Welch, and Daniell) reduce the variance of the periodogram at the expense of increasing the bias (or, equivalently, worsening the resolution). The FBA interpretation of these modified methods provides an interesting explanation of their behavior. In the filter-bank context, the basic idea behind all of these modified periodograms is *to improve the power-calculation stage*, which is done so poorly within the unmodified periodogram.

The Bartlett and Welch methods split the available sample in several subsequences which are separately (bandpass) filtered. In principle, the larger the number of subsequences, the more samples are averaged in the power-calculation stage and the smaller the variance of the estimated PSD, but the worse the resolution (owing to the inability to design an appropriately narrow bandpass filter for a small-aperture subsequence).

The Daniell method, on the other hand, does not split the sample of observations but processes it as a whole. This method improves the "power calculation" in a different way. For each value of $\phi(\omega)$ to be estimated, a number of different bandpass filters are employed, each with center frequency near ω. Each bandpass filter yields only one sample of the filtered signal, but, there being several bandpass filters, we might get enough information for the power-calculation stage. As the number of filters used increases, the variance of the estimated PSD decreases but the resolution becomes worse (since $\phi(\omega)$ is implicitly assumed to be constant over a wider and wider frequency interval centered on the current ω and approximately equal to the union of the filters' passbands).

5.3 REFINED FILTER-BANK METHOD

The bandpass filter used in the periodogram is only one of many possible choices. The periodogram was *not* designed as a filter-bank method, so we might well wonder whether we could find other, better choices of the bandpass filter. In this section, we present a refined filter bank (RFB) approach to spectral estimation. Such an approach was introduced in [Thomson 1982] and was further developed in [Mullis and Scharf 1991]. More recent references on this approach include [Bronez 1992; Onn and Steinhardt 1993; Riedel and Sidorenko 1995].

For the discussion that follows, it is convenient to use a *baseband filter* in the filter-bank approach of Figure 5.1, in lieu of the bandpass filter. Let $H_{\mathrm{BF}}(\omega)$ denote the frequency response of the bandpass filter with center frequency $\tilde{\omega}$ (say), and let the baseband filter be defined by

$$H(\omega) = H_{\mathrm{BF}}(\omega + \tilde{\omega}) \tag{5.3.1}$$

(the center frequency of $H(\omega)$ being equal to zero). If the input to the FBA scheme is also modified such that

$$\boxed{y(t) \longrightarrow \tilde{y}(t) = e^{-i\tilde{\omega}t} y(t)} \tag{5.3.2}$$

then, according to the complex (de)modulation formula (1.4.11), the output of the scheme is left unchanged by the translation in (5.3.1) of the passband down to baseband. In order to help interpret the previous transformations, we depict in Figure 5.3 the type of PSD translation implied by the *demodulation process* in (5.3.2). It is clearly seen from this figure that the problem of isolating the band around $\tilde{\omega}$ by bandpass filtering becomes one of baseband filtering. The modified FBA scheme is shown in Figure 5.4. The baseband-filter design problem is the subject of the next subsection.

5.3.1 Slepian Baseband Filters

In the following, we address the problem of designing a finite-impulse response (FIR) baseband filter that passes the *baseband*

$$[-\beta\pi, \beta\pi] \tag{5.3.3}$$

as undistorted as possible and that attenuates the frequencies outside baseband as much as possible. Let

$$h = [h_0 \dots h_{N-1}]^* \tag{5.3.4}$$

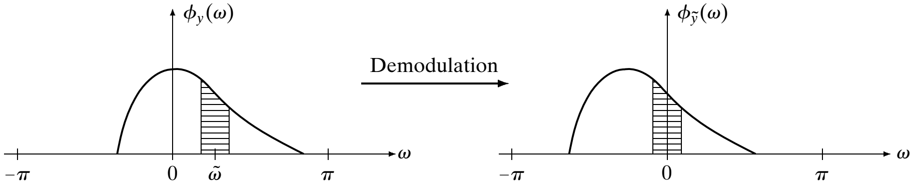


Figure 5.3 The relationship between the PSDs of the original signal $y(t)$ and the demodulated signal $\tilde{y}(t)$.

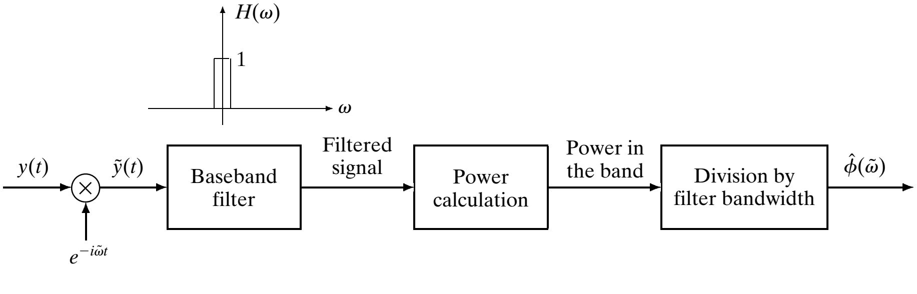


Figure 5.4 The modified filter-bank approach to PSD estimation.

denote the impulse response of such a filter, and let

$$H(\omega) = \sum_{k=0}^{N-1} h_k e^{-i\omega k} = h^* a(\omega)$$

(where $a(\omega) = [1 \; e^{-i\omega} \ldots \; e^{-i(N-1)\omega}]^T$) be the corresponding frequency response. The two design objectives can be turned into mathematical specifications in the following way: Let the input to the filter be *white noise* of unit variance. Then the power of the output is

$$\begin{aligned} \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 d\omega &= \sum_{k=0}^{N-1}\sum_{p=0}^{N-1} h_k h_p^* \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{i\omega(p-k)} d\omega\right] \\ &= \sum_{k=0}^{N-1}\sum_{p=0}^{N-1} h_k h_p^* \delta_{k,p} = h^* h \end{aligned} \qquad (5.3.5)$$

We note in passing that equation (5.3.5) can be recognized as Parseval's theorem (1.2.6). The part of the total power, (5.3.5), that lies in the baseband is given by

$$\frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} |H(\omega)|^2 d\omega = h^* \left\{\frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} a(\omega) a^*(\omega) d\omega\right\} h \triangleq h^* \Gamma h \qquad (5.3.6)$$

The k, p element of the $N \times N$ matrix Γ defined in (5.3.6) is given by

$$\Gamma_{k,p} = \frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} e^{-i(k-p)\omega} d\omega = \frac{\sin[(k-p)\beta\pi]}{(k-p)\pi} \qquad (5.3.7)$$

which, with the use of the sinc function, can be written as

$$\boxed{\Gamma_{k,p} = \beta \mathrm{sinc}[(k-p)\beta\pi] \triangleq \gamma_{|k-p|}} \qquad (5.3.8)$$

Note that the matrix Γ is symmetric and Toeplitz. Also, note that this matrix has already been encountered in the window design example in Section 2.6.3. In fact, as we will shortly see, the window-design strategy in that example is quite similar to the baseband-filter design method employed here.

Since the filter h must be such that the power of the filtered signal in the baseband is as large as possible relative to the total power, we are led to the following optimization problem:

$$\boxed{\max_h h^* \Gamma h \qquad \text{subject to} \quad h^* h = 1} \qquad (5.3.9)$$

The solution to the previous problem is given in Result R13 in Appendix A: the maximizing h is equal to the eigenvector of Γ corresponding to its maximum eigenvalue. Hence, we have proven the following result:

> The impulse response h of the "most selective" baseband filter (according to the design objectives in (5.3.9)) is given by the dominant eigenvector of Γ. (It is called the *first Slepian sequence*.) (5.3.10)

The matrix Γ played a key role in the foregoing derivation. In what follows, we look in more detail at the *eigenstructure of* Γ. In particular, we provide an intuitive explanation of why the first dominant eigenvector of Γ behaves like a baseband filter. We also show that, depending on the relation between β and N, the next dominant eigenvectors of Γ might also be used as baseband filters. Our discussion of these aspects will be partly heuristic. Note that the eigenvectors of Γ are called the *Slepian sequences* [SLEPIAN 1964] (as mentioned in (5.3.10)). We denote these eigenvectors by $\{s_k\}_{k=1}^N$.

Remark: The Slepian sequences should not be computed by the eigendecomposition of Γ. Numerically more efficient and reliable ways for computing these sequences exist (see, e.g., [SLEPIAN 1964]), for instance, as solutions to some differential equations or as eigenvectors of certain tridiagonal matrices. ■

The theoretical eigenanalysis of Γ is a difficult problem in the case of finite N. (Of course, the eigenvectors and eigenvalues of Γ can always be *computed*, for given β and N; here we are interested in establishing *theoretical expressions* for Γ's eigenelements.) For N sufficiently large, however, "reasonable approximations" to the eigenelements of Γ can be derived. Let $a(\omega)$ be defined as before:

$$a(\omega) = [1 \;\; e^{-i\omega} \ldots \; e^{-i(N-1)\omega}]^T \tag{5.3.11}$$

Assume that β is *chosen larger than* $1/N$, and define

$$\boxed{K = N\beta \geq 1} \tag{5.3.12}$$

(To simplify the discussion, K and N are assumed to be even integers in what follows.) With these preparations, and assuming that N is large, we can approximate the integral in (5.3.6) and write Γ as

$$\begin{aligned}\Gamma &\simeq \frac{1}{2\pi}\sum_{p=-K/2}^{K/2-1} a\left(\frac{2\pi}{N}p\right) a^*\left(\frac{2\pi}{N}p\right)\frac{2\pi}{N} \\ &= \frac{1}{N}\sum_{p=-K/2}^{K/2-1} a\left(\frac{2\pi}{N}p\right) a^*\left(\frac{2\pi}{N}p\right) \triangleq \Gamma_0\end{aligned} \tag{5.3.13}$$

The vectors $\{a(\frac{2\pi}{N}p)/\sqrt{N}\}_{p=-\frac{N}{2}+1}^{\frac{N}{2}}$, part of which appear in (5.3.13), can readily be shown to form an *orthonormal set*:

$$\frac{1}{N}a^*\left(\frac{2\pi}{N}p\right)a\left(\frac{2\pi}{N}s\right)=\frac{1}{N}\sum_{k=0}^{N-1}e^{i\frac{2\pi}{N}(p-s)k}$$
$$=\begin{cases}\dfrac{1}{N}\;\dfrac{e^{i2\pi(p-s)}-1}{e^{i\frac{2\pi}{N}(p-s)}-1}=0, & s\neq p\\[2ex] 1, & s=p\end{cases}\tag{5.3.14}$$

The eigenvectors of the matrix on the right-hand side of equation (5.3.13), Γ_0, are therefore given by $\{a\left(\frac{2\pi}{N}p\right)/\sqrt{N}\}_{p=-N/2+1}^{N/2}$, with eigenvalues of 1 (with multiplicity K) and 0 (with multiplicity $N-K$). The eigenvectors corresponding to the eigenvalues equal to one are $\{a\left(\frac{2\pi}{N}p\right)/\sqrt{N}\}_{p=-K/2+1}^{K/2}$. By paralleling the calculations in (5.2.3)–(5.2.5), it is not hard to show that each of these dominant eigenvectors of Γ_0 is the impulse response of a narrow bandpass filter with bandwidth equal to about $1/N$ and center frequency $\frac{2\pi}{N}p$; the set of these filters therefore covers the interval $[-\beta\pi,\beta\pi]$.

Now, the elements of Γ approach those of Γ_0 as N increases; more precisely, $|[\Gamma]_{i,j}-[\Gamma_0]_{i,j}|=\mathcal{O}(1/N)$ for sufficiently large N. However, this does *not* mean that $\|\Gamma-\Gamma_0\|\to 0$, as $N\to\infty$, for any reasonable matrix norm, because Γ and Γ_0 are $(N\times N)$ matrices. Consequently, the eigenelements of Γ do *not* necessarily converge to the eigenelements of Γ_0 as $N\to\infty$. However, given the previous analysis, we can at least expect that the eigenelements of Γ are not "too different" from those of Γ_0. This observation of the theoretical analysis, backed up with empirical evidence from the computation of the eigenelements of Γ in specific cases, leads us to conclude the following:

> The matrix Γ has K eigenvalues close to one and $(N-K)$ eigenvalues close to zero, provided N is large enough, where K is given by the "time-bandwidth" product (5.3.12). The dominant eigenvectors corresponding to the K largest eigenvalues form a set of orthogonal impulse responses of K bandpass filters that approximately cover the baseband $[-\beta\pi,\beta\pi]$. (5.3.15)

As we argue in the next subsections, in some situations (specified there) we may want to use the whole set of K *Slepian baseband filters*, not only the dominant Slepian filter in this set.

5.3.2 RFB Method for High-Resolution Spectral Analysis

Assume that the spectral analysis problem dealt with is one in which it is important to achieve the maximum resolution afforded by the approach at hand (such a problem appears, for instance, in the case of PSDs with closely spaced peaks). Then we set

$$\boxed{\beta=1/N\iff K=1}\tag{5.3.16}$$

(Note that we cannot set β to a value less than $1/N$. That choice would lead to $K < 1$, which is meaningless. The fact that we must choose $\beta \geq 1/N$ is one of the many facets of the $1/N$-resolution limit of the nonparametric spectral estimation.) Because $K = 1$, we cannot use more than the first Slepian sequence as a bandpass filter:

$$h = s_1 \tag{5.3.17}$$

The way in which the RFB scheme based on (5.3.17) works is described next.

First, note from (5.3.5), (5.3.9), and (5.3.16) that

$$\begin{aligned} 1 = h^*h &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 d\omega \simeq \frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi} |H(\omega)|^2 d\omega \\ &\simeq \beta |H(0)|^2 = \frac{1}{N}|H(0)|^2 \end{aligned} \tag{5.3.18}$$

Hence, under the (idealizing) assumption that $H(\omega)$ is different from zero only in the baseband where it takes a constant value, we have

$$|H(0)|^2 \simeq N \tag{5.3.19}$$

Next, consider the sample at the filter's output obtained by the convolution of the whole input sequence $\{\tilde{y}(t)\}_{t=1}^{N}$ with the filter impulse response $\{h_k\}$:

$$x \triangleq \sum_{k=0}^{N-1} h_k \tilde{y}(N-k) = \sum_{t=1}^{N} h_{N-t}\tilde{y}(t) \tag{5.3.20}$$

The power of x should be approximately equal to the PSD value $\phi(\tilde{\omega})$, as is confirmed by the following calculation:

$$\begin{aligned} E\left\{|x|^2\right\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \phi_{\tilde{y}}(\omega) d\omega \\ &\simeq \frac{N}{2\pi}\int_{-\beta\pi}^{\beta\pi} \phi_{\tilde{y}}(\omega) d\omega = \frac{N}{2\pi}\int_{-\beta\pi}^{\beta\pi} \phi_y(\omega + \tilde{\omega}) d\omega \\ &\simeq \frac{N}{2\pi}\phi_y(\tilde{\omega}) \times 2\pi\beta = N\beta\phi_y(\tilde{\omega}) = \phi_y(\tilde{\omega}) \end{aligned} \tag{5.3.21}$$

The second "equality" above follows from the properties of $H(\omega)$ (see, also, (5.3.19)), the third from the complex demodulation formula (1.4.11), and the fourth from the assumption that $\phi_y(\omega)$ is nearly constant over the passband considered.

In view of (5.3.21), the PSD estimation problem reduces to estimating the power of the filtered signal. Since only one sample, x, of that signal is available, the obvious estimate for the signal power is $|x|^2$. This leads to the following estimate of $\phi(\omega)$:

$$\boxed{\hat{\phi}(\omega) = \left| \sum_{t=1}^{N} h_{N-t} y(t) e^{-i\omega t} \right|^2} \tag{5.3.22}$$

Here $\{h_k\}$ is given by the first Slepian sequence (see (5.3.17)). The reason we did not divide (5.3.22) by the filter bandwidth is that $|H(0)|^2 \simeq N$, by (5.3.19), which differs from assumption (ii) in (5.1.5).

The spectral estimate (5.3.22) is recognized to be a *windowed periodogram* with *temporal window* $\{h_{N-k}\}$. For large values of N, it follows from the analysis in the previous section that h can be expected to be reasonably close to the vector $[1 \dots 1]^T/\sqrt{N}$. When inserting the latter vector in (5.3.22), we get the unwindowed periodogram. Hence, we reach the conclusion that, *for N large enough, the RFB estimate* (5.3.22) *will behave not too differently from the unmodified periodogram* (as is quite natural, in view of the fact that we wanted a high-resolution spectral estimator, and the basic periodogram is known to be such an estimator).

Remark: We warn the reader, once again, that the foregoing discussion is heuristic. As explained before (see the discussion related to (5.3.15)), as N increases, $\{h_k\}$ may be expected to be "reasonably close" to but not necessarily to converge to $1/\sqrt{N}$. In addition, even if $\{h_k\}$ in (5.3.22) converges to $1/\sqrt{N}$ as $N \to \infty$, the function in (5.3.22) might not converge to $\hat{\phi}_p(\omega)$ if the convergence *rate* of $\{h_k\}$ is too slow. (Note that the number of $\{h_k\}$ in (5.3.22) is equal to N.) Hence, $\hat{\phi}(\omega)$ in (5.3.22) and the periodogram $\hat{\phi}_p(\omega)$ could differ from one another even for large values of N. ■

In any case, even though the two estimators $\hat{\phi}(\omega)$ in (5.3.22) and $\hat{\phi}_p(\omega)$ generally give different PSD values, they both base the power-calculation stage of the FBA scheme on only a single sample. Hence, similarly to $\hat{\phi}_p(\omega)$, the RFB estimate (5.3.22) is expected to exhibit erratic fluctuations. The next subsection discusses a way in which the variance of the RFB spectral estimate can be reduced, at the expense of reducing the resolution of this estimate.

5.3.3 RFB Method for Statistically Stable Spectral Analysis

The FBA interpretation of the modified periodogram methods, as explained in Section 5.2, highlighted two approaches to reduce the statistical variability of the spectral estimate (5.3.22). The *first approach* consists of splitting the available sample $\{y(t)\}_{t=1}^{N}$ into a number of subsequences, computing (5.3.22) for each subsequence, and then averaging the values so obtained. The problem with this way of proceeding is that the values taken by (5.3.22) for different subsequences are not guaranteed to be statistically independent. In fact, if the subsequences overlap, then those values could be strongly correlated. The consequence of this fact is that one can never be sure of the "exact" reduction of variance that is achieved by averaging, in a given situation.

The *second approach* to reducing the variance consists of using several bandpass filters, in lieu of only one, which operate on the whole data sample [THOMSON 1982]. This approach aims at producing *statistically independent samples for the power-calculation stage*. When this is achieved, *the variance is reduced K times*, where K is the number of samples averaged (which equals the number of bandpass filters used).

In the following, we focus on this second approach, which appears particularly suitable for the RFB method. We set β to some value larger than $1/N$, which gives (*cf.* (5.3.12))

$$\boxed{K = N\beta > 1} \tag{5.3.23}$$

The larger β is (i.e., the lower is the resolution), the larger is K, and hence the larger is the reduction in variance that can be achieved. By using the result (5.3.15), we define K baseband filters as

$$\boxed{h_p = [h_{p,0} \dots h_{p,N-1}]^* = s_p, \qquad (p = 1, \dots, K)} \tag{5.3.24}$$

Here, h_p denotes the impulse response vector of the pth filter and s_p is the pth dominant Slepian sequence. Note that s_p is real valued (see Result R12 in Appendix A), and thus so is h_p. According to the discussion leading to (5.3.15), the set of filters (5.3.24) covers the baseband $[-\beta\pi, \beta\pi]$, with each of these filters passing (roughly speaking) $1/K$ of this baseband. Let x_p be defined similarly to x in (5.3.20), but now, for the pth filter,

$$x_p = \sum_{k=0}^{N-1} h_{p,k}\tilde{y}(N-k) = \sum_{t=1}^{N} h_{p,N-t}\tilde{y}(t) \tag{5.3.25}$$

The calculation (5.3.21) applies to $\{x_p\}$ in exactly the same way; hence,

$$E\left\{|x_p|^2\right\} \simeq \phi_y(\tilde{\omega}), \qquad p = 1, \dots, K \tag{5.3.26}$$

In addition, a straightforward calculation gives

$$\begin{aligned}
E\left\{x_p x_k^*\right\} &= E\left\{\left[\sum_{t=0}^{N-1} h_{p,t}\tilde{y}(N-t)\right]\left[\sum_{s=0}^{N-1} h_{k,s}^*\tilde{y}^*(N-s)\right]\right\} \\
&= \sum_{t=0}^{N-1}\sum_{s=0}^{N-1} h_{p,t}h_{k,s}^* r_{\tilde{y}}(s-t) \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{t=0}^{N-1}\sum_{s=0}^{N-1} h_{p,t}h_{k,s}^*\phi_{\tilde{y}}(\omega)e^{i(s-t)\omega}d\omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_p(\omega)H_k^*(\omega)\phi_{\tilde{y}}(\omega)d\omega
\end{aligned}$$

$$\begin{aligned} &\simeq \phi_{\tilde{y}}(0) h_p^* \left[\frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} a(\omega) a^*(\omega) d\omega \right] h_k \\ &= \phi_y(\tilde{\omega}) h_p^* \Gamma h_k = 0 \qquad \text{for } k \neq p \end{aligned} \tag{5.3.27}$$

Thus, the random variables x_p and x_k (for $p \neq k$) are approximately uncorrelated under the assumptions made. This implies, at least under the assumption that the $\{x_k\}$ are Gaussian, that $|x_p|^2$ and $|x_k|^2$ are *statistically independent* (for $p \neq k$).

According to the previous calculations, $\{|x_p|^2\}_{p=1}^K$ can approximately be considered to be independent random variables all with the same mean $\phi_y(\tilde{\omega})$. Then we can estimate $\phi_y(\tilde{\omega})$ by the following average of $\{|x_p|^2\}$: $\frac{1}{K} \sum_{p=1}^K |x_p|^2$, or

$$\boxed{\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} h_{p,N-t} y(t) e^{-i\omega t} \right|^2} \tag{5.3.28}$$

We may suspect that the random variables $\{|x_p|^2\}$ have not only the same mean, but also the same variance (this can, in fact, be readily shown under the Gaussian hypothesis). Whenever this is true, the variance of the average in (5.3.28) is K times smaller than the variance of each of the variables averaged. The above findings are summarized in the following:

If the resolution threshold β is increased K times from $\beta = 1/N$ (the lowest value) to $\beta = K/N$, then the variance of the RFB estimate in (5.3.22) may be reduced by a factor K by constructing the spectral estimate as in (5.3.28), where the pth baseband filter's impulse response $\{h_{p,t}\}_{t=0}^{N-1}$ is given by the pth dominant Slepian sequence ($p = 1, \ldots, K$). (5.3.29)

The RFB spectral estimator (5.3.28) can be given two interpretations. First, arguments similar with those following equation (5.3.22) suggest that, *for large N, the RFB estimate (5.3.28) behaves similarly to the Daniell method of periodogram averaging*. For small or medium-sized values of N, the RFB and Daniell methods behave differently. In such a case, we can relate (5.3.28) to the class of *multiwindow spectral estimators* [Thomson 1982]. Indeed, the RFB estimate (5.3.28) can be interpreted as the average of K windowed periodograms, where the pth periodogram is computed from the raw data sequence $\{y(t)\}$ windowed with the pth dominant Slepian sequence. Note that the Slepian sequences are given by the eigenvectors of the real *Toeplitz* matrix Γ, so they must be either symmetric ($h_{p,N-t} = h_{p,t-1}$) or skew–symmetric ($h_{p,N-t} = -h_{p,t-1}$)—see Result R25 in Appendix A. This means that (5.3.28) can be written alternatively, as

$$\boxed{\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} h_{p,t-1} y(t) e^{-i\omega t} \right|^2} \tag{5.3.30}$$

This form of the RFB estimate makes its interpretation as a multiwindow spectrum estimator more direct.

For a second interpretation of the RFB estimate (5.3.28), consider the (Daniell-type) spectrally smoothed periodogram estimator

$$\begin{aligned}
\hat{\phi}(\tilde{\omega}) &= \frac{1}{2\pi\beta}\int_{\tilde{\omega}-\beta\pi}^{\tilde{\omega}+\beta\pi}\hat{\phi}_p(\omega)d\omega = \frac{1}{2\pi\beta}\int_{-\beta\pi}^{\beta\pi}\hat{\phi}_p(\omega+\tilde{\omega})d\omega \\
&= \frac{1}{2\pi\beta}\int_{-\beta\pi}^{\beta\pi}\frac{1}{N}\left|\sum_{t=1}^{N}y(t)e^{-i(\omega+\tilde{\omega})t}\right|^2 d\omega \\
&= \frac{1}{2\pi K}\int_{-\beta\pi}^{\beta\pi}\sum_{t=1}^{N}\sum_{s=1}^{N}\tilde{y}(t)\tilde{y}^*(s)e^{-i\omega t}e^{i\omega s}d\omega \\
&= \frac{1}{K}[\tilde{y}^*(1)\ \dots\ \tilde{y}^*(N)] \\
&\quad \cdot\left\{\frac{1}{2\pi}\int_{-\beta\pi}^{\beta\pi}\begin{bmatrix}1\\ e^{i\omega}\\ \vdots\\ e^{i(N-1)\omega}\end{bmatrix}[1\ \ e^{-i\omega}\dots e^{-i(N-1)\omega}]\,d\omega\right\}\begin{bmatrix}\tilde{y}(1)\\ \vdots\\ \tilde{y}(N)\end{bmatrix} \\
&= \frac{1}{K}[\tilde{y}^*(1)\ \dots\ \tilde{y}^*(N)]\Gamma\begin{bmatrix}\tilde{y}(1)\\ \vdots\\ \tilde{y}(N)\end{bmatrix}
\end{aligned} \tag{5.3.31}$$

where we made use of the fact that Γ is real valued. It follows from the result (5.3.15) that Γ can be approximated by the *rank-K* matrix

$$\Gamma \simeq \sum_{p=1}^{K} s_p s_p^T = \sum_{p=1}^{K} h_p h_p^T \tag{5.3.32}$$

Inserting (5.3.32) into (5.3.31) and using the fact that the Slepian sequences $s_p = h_p$ are real valued leads to the PSD estimator

$$\hat{\phi}(\tilde{\omega}) \simeq \frac{1}{K}\sum_{p=1}^{K}\left|\sum_{t=1}^{N} h_{p,t-1}\tilde{y}(t)\right|^2 \tag{5.3.33}$$

which is precisely the RFB estimator (5.3.30). Hence, the RFB estimate of the PSD can also be interpreted as a *reduced-rank smoothed periodogram.*

We might think of using the full-rank smoothed periodogram (5.3.31) as an estimator for PSD, in lieu of the reduced-rank smoothed periodogram (5.3.33) that coincides with the RFB estimate. However, from a theoretical standpoint, we have no strong reason to do so. Moreover, from a practical standpoint, we have clear reasons against such an idea. We can explain this briefly as follows: The K dominant eigenvectors of Γ can be *precomputed* with satisfactory numerical accuracy. Then, evaluation of (5.3.33) can be done by using an FFT algorithm in approximately $\frac{1}{2}KN\log_2 N = \frac{1}{2}\beta N^2 \log_2 N$ flops. On the other hand, a direct evaluation of (5.3.31) would require N^2 flops for each value of ω, which leads to a prohibitively large total computational burden. A computationally efficient evaluation of (5.3.31) would require some factorization of Γ to be performed, such as the eigendecomposition of Γ. However, Γ is an extremely ill-conditioned matrix (recall that $N-K = N(1-\beta)$ of its eigenvalues are close to zero), which means that such a complete factorization cannot easily be performed with satisfactory numerical accuracy. In any case, even if we were able to precompute the eigendecomposition of Γ, evaluation of (5.3.31) would require $\frac{1}{2}N^2\log_2 N$ flops, which is still larger by a factor of $1/\beta$ than what is required for (5.3.33).

5.4 CAPON METHOD

The periodogram was previously shown to be a filter-bank approach that uses a bandpass filter whose impulse response vector is given by the standard Fourier transform vector (i.e., $[1, e^{-i\tilde{\omega}}, \dots, e^{-i(N-1)\tilde{\omega}}]^T$). In the periodogram approach, there is *no attempt to purposely design* the bandpass filter to achieve some desired characteristics (see, however, Section 5.5). The RFB method, on the other hand, uses a bandpass filter specifically designed to be "*as selective as possible*" *for a white-noise input*. (See (5.3.5) and the discussion preceding it.) The RFB's filter is still *data independent* in the sense that it does not adapt to the processed data in any way. Presumably, it might be valuable to take the data properties into consideration when designing the bandpass filter. In other words, the filter should be designed to be "as selective as possible" (according to a criterion to be specified), not for a fictitious white-noise input, but for the input consisting of the studied data themselves. This is the basic idea behind the Capon method, which is an FBA procedure based on a *data-dependent bandpass filter* [Capon 1969; Lacoss 1971].

5.4.1 Derivation of the Capon Method

The Capon method (CM), in contrast to the RFB estimator (5.3.28), uses only *one bandpass filter* for computing one estimated spectrum value. This suggests that, if the CM is to provide statistically stable spectral estimates, then it should make use of the other approach which affords this: *splitting the raw sample into subsequences* and averaging the results obtained from each subsequence. Indeed, as we shall see, the Capon method is based essentially on this second approach.

Consider a filter with a finite impulse response of length m, denoted by

$$h = [h_0 \ \ h_1 \dots h_m]^* \tag{5.4.1}$$

where m is a positive integer that is unspecified for the moment. The output of the filter at time t, when the input is the raw data sequence $\{y(t)\}$, is given by

$$\begin{aligned} y_F(t) &= \sum_{k=0}^{m} h_k y(t-k) \\ &= h^* \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \end{aligned} \tag{5.4.2}$$

Let R denote the covariance matrix of the data vector in (5.4.2). Then the power of the filter output can be written as

$$E\left\{|y_F(t)|^2\right\} = h^*Rh \tag{5.4.3}$$

where, according to the preceding definition,

$$R = E\left\{ \begin{bmatrix} y(t) \\ \vdots \\ y(t-m) \end{bmatrix} [y^*(t) \ldots y^*(t-m)] \right\} \tag{5.4.4}$$

The response of the filter (5.4.2) to a sinusoidal component of frequency ω (say) is determined by the filter's frequency response—that is,

$$H(\omega) = \sum_{k=0}^{m} h_k e^{-i\omega k} = h^* a(\omega) \tag{5.4.5}$$

where

$$a(\omega) = [1 \quad e^{-i\omega} \ldots e^{-im\omega}]^T \tag{5.4.6}$$

If we want to make the filter as selective as possible for a frequency band around the current value ω, then we may think of minimizing the total power in (5.4.3), subject to the constraint that the filter pass the frequency ω undistorted. This idea leads to the following optimization problem:

$$\boxed{\min_h h^*Rh \quad \text{subject to } h^*a(\omega) = 1} \tag{5.4.7}$$

The solution to (5.4.7) is given in Result R35 in Appendix A:

$$\boxed{h = R^{-1}a(\omega)/a^*(\omega)R^{-1}a(\omega)} \tag{5.4.8}$$

Inserting (5.4.8) into (5.4.3) gives

$$E\left\{|y_F(t)|^2\right\} = 1/a^*(\omega)R^{-1}a(\omega) \tag{5.4.9}$$

This is the power of $y(t)$ in a passband centered on ω. Then, assuming that the (idealized) conditions (i) and (ii) in (5.1.5) hold, we can compute the value of the PSD of $y(t)$ at the passband's center frequency as

$$\phi(\omega) \simeq \frac{E\left\{|y_F(t)|^2\right\}}{\beta} = \frac{1}{\beta a^*(\omega)R^{-1}a(\omega)} \tag{5.4.10}$$

where β denotes the frequency bandwidth of the filter given by (5.4.8). The division by β is sometimes omitted in the literature, but it is required to complete the FBA scheme in Figure 5.1. Note that, since the bandpass filter (5.4.8) is data dependent, its bandwidth β is not necessarily data independent, nor is it necessarily frequency independent. Hence, the division by β in (5.4.10) might fail to represent a simple scaling of $E\left\{|y_F(t)|^2\right\}$; it could change the shape of this quantity as a function of ω.

There are various possibilities for choosing the bandwidth β, depending on the degree of precision we are aiming for. The simplest possibility is to set

$$\beta = 1/(m+1) \tag{5.4.11}$$

This choice is motivated by the time-bandwidth product result (2.6.5), which says that, for a filter whose temporal aperture is equal to $(m+1)$, the bandwidth should be given roughly by $1/(m+1)$. By inserting (5.4.11) in (5.4.10), we obtain

$$\phi(\omega) \simeq \frac{(m+1)}{a^*(\omega)R^{-1}a(\omega)} \tag{5.4.12}$$

Note that, if $y(t)$ is white noise of variance σ^2, (5.4.12) takes the correct value: $\phi(\omega) = \sigma^2$. In the general case, however, (5.4.11) gives only a rough indication of the filter's bandwidth, because the time-bandwidth product result does not apply exactly to the present situation. (See the conditions under which (2.6.5) was derived.)

An often more exact expression for β can be obtained as follows [Lagunas, Santamaria, Gasull, and Moreno 1986]: The (equivalent) bandwidth of a bandpass filter can be defined as the support of the rectangle centered on ω (the filter's center frequency) that concentrates the whole energy in the filter's frequency response. According to this definition, β can be assumed to satisfy

$$\int_{-\pi}^{\pi} |H(\psi)|^2 d\psi = |H(\omega)|^2 2\pi\beta \tag{5.4.13}$$

In the present case, $H(\omega) = 1$ (see (5.4.7)), so we obtain from (5.4.13)

$$\beta = \frac{1}{2\pi}\int_{-\pi}^{\pi} |h^* a(\psi)|^2 d\psi = h^*\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} a(\psi)a^*(\psi)d\psi\right]h \tag{5.4.14}$$

The (k, p) element of the central matrix in the preceding quadratic form is given by

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-i\psi(k-p)} d\psi = \delta_{k,p} \tag{5.4.15}$$

With this observation and (5.4.8), (5.4.14) leads to

$$\beta = h^* h = \frac{a^*(\omega) R^{-2} a(\omega)}{[a^*(\omega) R^{-1} a(\omega)]^2} \tag{5.4.16}$$

Note that this expression of the bandwidth is both data and frequency dependent (as was alluded to previously). Inserting (5.4.16) in (5.4.10) gives

$$\phi(\omega) \simeq \frac{a^*(\omega) R^{-1} a(\omega)}{a^*(\omega) R^{-2} a(\omega)} \tag{5.4.17}$$

Remark: The expression for β in (5.4.16) is based on the assumption that most of the area under the curve of $|H(\psi)|^2 = |h^* a(\psi)|^2$ (for $\psi \in [-\pi, \pi]$) is located around the center frequency ω. This assumption is true often, but not always. For instance, consider a data sequence $\{y(t)\}$ consisting of a number of sinusoidal components with frequencies $\{\omega_k\}$ in noise with *small* power. Then the Capon filter (5.4.8) with center frequency ω will likely place nulls at $\{\psi = \omega_k\}$ to annihilate the strong sinusoidal components in the data, but will pay little attention to the weak noise component. The consequence is that $|H(\psi)|^2$ will be nearly zero at $\{\psi = \omega_k\}$, and nearly 1 at $\psi = \omega$ (by (5.4.7)), but may take rather large values at other frequencies (see, for example, the numerical examples in [Li and Stoica 1996a], which demonstrate this behavior of the Capon filter). In such a case, the formula (5.4.16) could significantly overestimate the "true" bandwidth; hence, the spectral formula (5.4.17) could significantly underestimate the PSD $\phi(\omega)$. ■

In the derivations above, the true data-covariance matrix R has been assumed available. In order to turn the previous PSD formulas into practical spectral estimation algorithms, we must replace R in these formulas by a sample estimate—for instance, by

$$\hat{R} = \frac{1}{N-m} \sum_{t=m+1}^{N} \begin{bmatrix} y(t) \\ \vdots \\ y(t-m) \end{bmatrix} [y^*(t) \ldots y^*(t-m)] \tag{5.4.18}$$

Doing so, we obtain the following two spectral estimators corresponding to (5.4.12) and (5.4.17), respectively:

$$\boxed{\text{CM-Version 1:} \quad \hat{\phi}(\omega) = \frac{m+1}{a^*(\omega) \hat{R}^{-1} a(\omega)}} \tag{5.4.19}$$

$$\boxed{\text{CM-Version 2:} \quad \hat{\phi}(\omega) = \frac{a^*(\omega) \hat{R}^{-1} a(\omega)}{a^*(\omega) \hat{R}^{-2} a(\omega)}} \tag{5.4.20}$$

There is an implicit assumption in both (5.4.19) and (5.4.20) that $\hat{R}^{-1}$ exists. This assumption sets a limit on the maximum value that can be chosen for m:

$$\boxed{m < N/2} \qquad (5.4.21)$$

(Observe that $\text{rank}(\hat{R}) \leq N - m$, which is less than $\dim(\hat{R}) = m + 1$ if (5.4.21), is violated.) The inequality (5.4.21) is important, since it sets a limit on the resolution achievable by the Capon method. Indeed, since the Capon method is based on a bandpass filter with impulse response's aperture equal to m, we may expect its resolution threshold to be on the order of $1/m > 2/N$ (with the inequality following from (5.4.21)).

As m is decreased, we can expect the resolution of Capon method to become worse (*cf.* the previous discussion). On the other hand, the accuracy with which $\hat{R}$ is estimated increases with decreasing m, because more outer products are averaged in (5.4.18). The main consequence of the increased accuracy of $\hat{R}$ is to statistically stabilize the spectral estimate (5.4.19) or (5.4.20). Hence, the choice of m should be made with the ubiquitous tradeoff between resolution and statistical accuracy in mind. It is interesting to note that, for the Capon method, both the filter-design and the power-calculation stages are data dependent. The accuracy of both these stages could worsen if m is chosen too large. In applications, the maximum value that can be chosen for m might also be limited by considerations of computational complexity.

Empirical studies have shown that *the ability of the Capon method to resolve fine details of a PSD, such as closely spaced peaks, is superior to the corresponding performance of the periodogram-based methods*. This superiority may be attributed to the higher statistical stability of Capon method, as explained next. For m smaller than $N/2$ (see (5.4.21)), we may expect the Capon method to possess worse resolution but better statistical accuracy compared with the unwindowed or "mildly windowed" periodogram method. It should be stressed that *the notion of "resolution" refers to the ability of the theoretically averaged spectral estimate $E\{\hat{\phi}(\omega)\}$ to resolve fine details in the true PSD $\phi(\omega)$*. This resolution is roughly inversely proportional to the window's length or the bandpass-filter impulse-response's aperture. *The "resolving power" corresponding to the estimate $\hat{\phi}(\omega)$ is more difficult to quantify*, but—of course—it is what interests us the most. It should be clear that the resolving power of $\hat{\phi}(\omega)$ depends not only on the bias of this estimate (i.e., on $E\{\hat{\phi}(\omega)\}$), but also on its variance. *A spectral estimator with low bias-based resolution, but high statistical accuracy may be better able to resolve fine details in a studied PSD than can a high resolution/low accuracy estimator.* Since the periodogram may achieve better bias-based resolution than the Capon method, the higher (empirically observed) "resolving power" of the Capon method should be due to a better statistical accuracy (i.e., a lower variance).

In the context of the previous discussion, it is interesting to note that the Blackman–Tukey periodogram with a Bartlett window of length $2m + 1$, which is given (see (2.5.1)) by

$$\hat{\phi}_{\text{BT}}(\omega) = \sum_{k=-m}^{m} \frac{(m + 1 - |k|)}{m + 1} \hat{r}(k) e^{-i\omega k}$$

can be written in a form that bears some resemblance to the form (5.4.19) of the CM-Version 1 estimator. A straightforward calculation gives

$$\hat{\phi}_{\mathrm{BT}}(\omega) = \sum_{t=0}^{m}\sum_{s=0}^{m} \hat{r}(t-s)e^{-i\omega(t-s)}/(m+1) \tag{5.4.22}$$

$$= \frac{1}{m+1}a^*(\omega)\hat{R}a(\omega) \tag{5.4.23}$$

where $a(\omega)$ is as defined in (5.4.6), and $\hat{R}$ is the Hermitian, Toeplitz sample covariance matrix

$$\hat{R} = \begin{bmatrix} \hat{r}(0) & \hat{r}(1) & \dots & \hat{r}(m) \\ \hat{r}^*(1) & \hat{r}(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{r}(1) \\ \hat{r}^*(m) & \dots & \hat{r}^*(1) & \hat{r}(0) \end{bmatrix}$$

Comparing the previous expression for $\hat{\phi}_{\mathrm{BT}}(\omega)$ with (5.4.19), it is seen that the *CM-Version 1 can be obtained from Blackman–Tukey estimator by replacing $\hat{R}$ in the Blackman–Tukey estimator with $\hat{R}^{-1}$ and then inverting the resultant quadratic form*. Next, we provide a brief explanation of why this replacement and inversion make sense. That is, if we ignore for a moment the technically sound filter-bank derivation of the Capon method, then why should the preceding way of obtaining CM-Version 1 from the Blackman–Tukey method provide a reasonable spectral estimator? We begin by noting (*cf.* Section 1.3.2) that

$$\lim_{m\to\infty} E\left\{ \frac{1}{m+1}\left|\sum_{t=0}^{m} y(t)e^{-i\omega t}\right|^2 \right\} = \phi(\omega)$$

However, a simple calculation shows that

$$E\left\{ \frac{1}{m+1}\left|\sum_{t=0}^{m} y(t)e^{-i\omega t}\right|^2 \right\} = \frac{1}{m+1}\sum_{t=0}^{m}\sum_{s=0}^{m} r(t-s)e^{-i\omega t}e^{i\omega s} = \frac{1}{m+1}a^*(\omega)Ra(\omega)$$

Hence,

$$\lim_{m\to\infty} \frac{1}{m+1}a^*(\omega)Ra(\omega) = \phi(\omega) \tag{5.4.24}$$

Similarly, one can show (see, e.g., [Hannan and Wahlberg 1989]) that

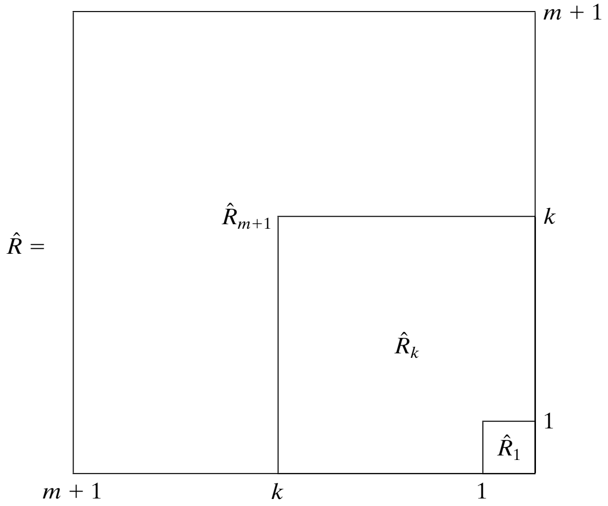
$$\lim_{m\to\infty} \frac{1}{m+1}a^*(\omega)R^{-1}a(\omega) = \phi^{-1}(\omega) \tag{5.4.25}$$

Comparing (5.4.24) with (5.4.25) provides the explanation we were looking for. Observe that the CM-Version 1 estimator is a finite-sample approximation to equation (5.4.25), whereas the Blackman–Tukey estimator is a finite-sample approximation to equation (5.4.24).

The Capon method has also been compared with the AR method of spectral estimation. (See Section 3.2.) It has been empirically observed that *the CM-Version 1 possesses less variance, but worse resolution than the AR spectral estimator*. This may be explained by making use of the relationship that exists between the CM-Version 1 and AR spectral estimators; see the next subsection (and also [Burg 1972]). The CM-Version 2 spectral estimator is less well studied; hence, its properties are not as well understood. In the next subsection, we also relate the CM-Version 2 to the AR spectral estimator. In the case of CM-Version 2, the relationship is more involved and so leaves less room for intuitive explanations.

5.4.2 Relationship between Capon and AR Methods

The AR method of spectral estimation was described in Chapter 3. Subsequently, we consider the covariance matrix estimate in (5.4.18). The AR method corresponding to this sample covariance matrix is the LS method discussed in Section 3.4.2. Let us denote the matrix $\hat{R}$ in (5.4.18) by $\hat{R}_{m+1}$, and its principal lower right $k \times k$ block by $\hat{R}_k$ $(k = 1, \ldots, m+1)$, as shown here:



(5.4.26)

With this notation, the coefficient vector θ_k and the residual power σ_k^2 of the *kth-order AR model* fitted to the data $\{y(t)\}$ are obtained as the solutions to the following matrix equation (refer to (3.4.6)):

$$\hat{R}_{k+1} \begin{bmatrix} 1 \\ \hat{\theta}_k^c \end{bmatrix} = \begin{bmatrix} \hat{\sigma}_k^2 \\ 0 \end{bmatrix} \tag{5.4.27}$$

(The complex conjugate in (5.4.27) appears because $\hat{R}_k$ is equal to the complex conjugate of the sample covariance matrix used in Chapter 3.) The nested structure of (5.4.26), along with the defining equation (5.4.27), implies

$$\hat{R}_{m+1}\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ & 1 & & \vdots & \vdots \\ & & \ddots & 0 & \\ & & & 1 & 0 \\ \hat{\theta}^c_m & \hat{\theta}^c_{m-1} & & \hat{\theta}^c_1 & 1 \end{bmatrix} = \begin{bmatrix} \hat{\sigma}^2_m & x & \dots & x \\ 0 & \hat{\sigma}^2_{m-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & x \\ 0 & \cdots & 0 & \hat{\sigma}^2_0 \end{bmatrix} \tag{5.4.28}$$

where "x" stands for undetermined elements. Let

$$\hat{\mathcal{H}} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ & 1 & & \vdots & \vdots \\ & & \ddots & 0 & \\ & & & 1 & 0 \\ \hat{\theta}^c_m & \hat{\theta}^c_{m-1} & & \hat{\theta}^c_1 & 1 \end{bmatrix} \tag{5.4.29}$$

It follows from (5.4.28) that

$$\hat{\mathcal{H}}^*\hat{R}_{m+1}\hat{\mathcal{H}} = \begin{bmatrix} \hat{\sigma}^2_m & x & \dots & x \\ & \hat{\sigma}^2_{m-1} & \ddots & \vdots \\ 0 & & \ddots & x \\ & & & \hat{\sigma}^2_0 \end{bmatrix} \tag{5.4.30}$$

(where, once more, x denotes undetermined elements). $\hat{\mathcal{H}}^*\hat{R}_{m+1}\hat{\mathcal{H}}$ is a Hermitian matrix, so the elements designated by "x" in (5.4.30) must be equal to zero. Hence, we have proven the following result, which is essential in establishing a relation between the AR and Capon methods of spectral estimation (and extends the one in Exercise 3.7 to the non-Toeplitz covariance case):

The parameters $\{\hat{\theta}_k, \hat{\sigma}^2_k\}$ of the AR models of orders $k = 1, 2, \dots, m$ determine the following factorization of the inverse (sample) covariance matrix:

$$\hat{R}^{-1}_{m+1} = \hat{\mathcal{H}}\hat{\Sigma}^{-1}\hat{\mathcal{H}}^* \quad ; \quad \hat{\Sigma} = \begin{bmatrix} \hat{\sigma}^2_m & & & 0 \\ & \hat{\sigma}^2_{m-1} & & \\ & & \ddots & \\ 0 & & & \hat{\sigma}^2_0 \end{bmatrix} \tag{5.4.31}$$

Let

$$\hat{A}_k(\omega) = [1 \quad e^{-i\omega} \ldots e^{-ik\omega}] \begin{bmatrix} 1 \\ \hat{\theta}_k \end{bmatrix} \tag{5.4.32}$$

denote the polynomial corresponding to the kth-order AR model, and let

$$\hat{\phi}_k^{\text{AR}}(\omega) = \frac{\hat{\sigma}_k^2}{|\hat{A}_k(\omega)|^2} \tag{5.4.33}$$

denote its associated PSD (see Chapter 3). It is readily verified that

$$\begin{aligned} a^*(\omega)\hat{\mathcal{H}} &= [1 \quad e^{i\omega} \ldots e^{im\omega}] \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ & 1 & & \vdots & \vdots \\ & & \ddots & 0 & \\ & & & 1 & 0 \\ \hat{\theta}_m^c & \hat{\theta}_{m-1}^c & & \hat{\theta}_1^c & 1 \end{bmatrix} \\ &= [\hat{A}_m^*(\omega), \ e^{i\omega}\hat{A}_{m-1}^*(\omega), \ \ldots, \ e^{im\omega}\hat{A}_0^*(\omega)] \end{aligned} \tag{5.4.34}$$

It follows from (5.4.31) and (5.4.34) that the quadratic form in the denominator of the CM–Version 1 spectral estimator can be written as

$$\begin{aligned} a^*(\omega)\hat{R}^{-1}a(\omega) &= a^*(\omega)\hat{\mathcal{H}}\hat{\Sigma}^{-1}\hat{\mathcal{H}}^*a(\omega) \\ &= \sum_{k=0}^{m} |\hat{A}_k(\omega)|^2/\hat{\sigma}_k^2 = \sum_{k=0}^{m} 1/\hat{\phi}_k^{\text{AR}}(\omega) \end{aligned} \tag{5.4.35}$$

which leads at once to

$$\boxed{\hat{\phi}_{\text{CM–1}}(\omega) = \frac{1}{\dfrac{1}{m+1}\displaystyle\sum_{k=0}^{m} 1/\hat{\phi}_k^{\text{AR}}(\omega)}} \tag{5.4.36}$$

which is the desired relation between the CM-Version 1 and the AR spectral estimates. This relation says that the inverse of the CM-Version 1 spectral estimator can be obtained by averaging the inverse estimated AR spectra of orders from 0 to m. In view of the averaging operation in (5.4.36), it is not difficult to understand why the CM-Version 1 possesses less statistical variability than the AR estimator. Moreover, the fact that the CM-Version 1 has also been found to have worse resolution and bias properties than the AR spectral estimate should be due to the presence of low-order AR models in (5.4.36).

Next, consider the CM-Version 2. The previous analysis of CM-Version 1 already provides a relation between the numerator in the spectral estimate corresponding to CM-Version 2, (5.4.20), and the AR spectra. In order to obtain a similar expression for the denominator in (5.4.20), some preparations are required. The (sample) covariance matrix $\hat{R}$ can be used to define $m+1$ *AR models of order* m, depending on which coefficient of the AR equation

$$\hat{a}_0 y(t) + \hat{a}_1 y(t-1) + \ldots + \hat{a}_m y(t-m) = \text{residuals} \tag{5.4.37}$$

we choose to set to 1. The AR model $\{\hat{\theta}_m, \hat{\sigma}_m^2\}$ used in the previous analysis corresponds to setting $\hat{a}_0 = 1$ in (5.4.37). However, in principle, any other AR coefficient in (5.4.37) may be normalized to one. The mth-order LS AR model obtained by setting $\hat{a}_k = 1$ in (5.4.37) is denoted by $\{\hat{\mu}_k =$ coefficient vector and $\hat{\gamma}_k =$ residual variance$\}$ and is given by the solution to the linear system of equations (compare with (5.4.27))

$$\hat{R}_{m+1}\hat{\mu}_k^c = \hat{\gamma}_k u_k \tag{5.4.38}$$

where the $(k+1)$st component of $\hat{\mu}_k$ is equal to one ($k = 0, \ldots, m$) and where u_k stands for the $(k+1)$st column of the $(m+1) \times (m+1)$ identity matrix:

$$u_k = [\underbrace{0 \ldots 0}_{k} \; 1 \; \underbrace{0 \ldots 0}_{m-k}]^T \tag{5.4.39}$$

Evidently, $[1 \; \hat{\theta}_m^T]^T = \hat{\mu}_0$ and $\hat{\sigma}_m^2 = \hat{\gamma}_0$.

As with (5.4.32) and (5.4.33), the (estimated) PSD corresponding to the kth mth-order AR model given by (5.4.38) is obtained as

$$\hat{\phi}_k^{\mathrm{AR}(m)}(\omega) = \frac{\hat{\gamma}_k}{|a^*(\omega)\hat{\mu}_k^c|^2} \tag{5.4.40}$$

It is shown, in the following calculation, that the denominator in (5.4.20) can be expressed as a (weighted) average of the AR spectra in (5.4.40):

$$\begin{aligned}\sum_{k=0}^{m} \frac{1}{\hat{\gamma}_k \hat{\phi}_k^{\mathrm{AR}(m)}(\omega)} &= \sum_{k=0}^{m} \frac{|a^*(\omega)\hat{\mu}_k^c|^2}{\hat{\gamma}_k^2} = a^*(\omega)\left[\sum_{k=0}^{m} \frac{\hat{\mu}_k^c \hat{\mu}_k^T}{\hat{\gamma}_k^2}\right] a(\omega) \\ &= a^*(\omega)\hat{R}^{-1}\left[\sum_{k=0}^{m} u_k u_k^*\right]\hat{R}^{-1} a(\omega) = a^*(\omega)\hat{R}^{-2} a(\omega)\end{aligned} \tag{5.4.41}$$

Combining (5.4.35) and (5.4.41) gives

$$\boxed{\hat{\phi}_{\mathrm{CM\text{-}2}}(\omega) = \frac{\sum_{k=0}^{m} 1/\hat{\phi}_k^{\mathrm{AR}}(\omega)}{\sum_{k=0}^{m} 1/\hat{\gamma}_k \hat{\phi}_k^{\mathrm{AR}(m)}(\omega)}} \tag{5.4.42}$$

This relation appears to be more involved, and hence more difficult to interpret, than the similar relation (5.4.36) corresponding to CM-Version 1. Nevertheless, (5.4.42) is still obtained by averaging various AR spectra, so we may reasonably expect that *the CM-Version 2 estimator, like the CM-Version 1 estimator, is more statistically stable but has poorer resolution than the AR spectral estimator*.

5.5 FILTER-BANK REINTERPRETATION OF THE PERIODOGRAM

As we saw in Section 5.2, the basic periodogram spectral estimator can be interpreted as an FBA method with a *preimposed* bandpass filter (whose impulse response is equal to the Fourier transform vector). In contrast, RFB and Capon are FBA methods based on *designed* bandpass filters. The filter used in the RFB method is data independent, whereas the filter is a function of the data covariances in the Capon method. The use of a data-dependent bandpass filter, such as in the Capon method, is intuitively appealing but it also leads to the following drawback: we need to obtain a consistent estimate of the filter impulse response, so the temporal aperture of the filter needs to be chosen (much) smaller than the sample length. This constraint sets a rather firm limit on the achievable spectral resolution. In addition, it appears that any filter-design methodology other than one originally suggested by Capon will most likely lead to a problem (such as an eigenanalysis) that should be solved for each value of the center frequency; doing so would be a rather prohibitive computational task. With these difficulties of the data-dependent design in mind, we may content ourselves with a “well-designed” data-independent filter. The purpose of this section is to show that *the basic periodogram and the Daniell method can be interpreted as FBA methods based on well-designed data-independent filters*, similar to the RFB method. As we will see, the bandpass filters used by the aforementioned periodogram methods are obtained *by combining the design procedures employed in the RFB and Capon methods*.

The next result is required. (See R35 in Appendix A for a proof.) Let R, H, A, and C be matrices of dimensions $(m \times m)$, $(m \times K)$, $(m \times n)$, and $(K \times n)$, respectively. Assume that R is positive definite and A has full column rank equal to n (and so $m \geq n$). Then the solution to the quadratic optimization problem with linear constraints,

$$\min_{H}(H^*RH) \quad \text{subject to} \quad H^*A = C$$

is given by

$$H = R^{-1}A(A^*R^{-1}A)^{-1}C^* \tag{5.5.1}$$

We can now proceed to derive our “new” FBA-based spectral estimation method. (As we will see shortly, it turns out that this method is not really new!) We would like this method to possess a facility for compromising between the bias and variance of the estimated PSD. As explained in the previous sections of this chapter, there are two main ways of doing this within the FBA: we either (i) use a bandpass filter with temporal aperture less than N, obtain the allowed number of samples of the filtered signal, and then calculate the power from these samples; or (ii) use a set of K bandpass filters with length-N impulse responses that cover a band centered on the current frequency value, obtain one sample of the filtered signals for each filter in the set, and calculate the power by averaging these K samples. As argued in Section 5.3, approach (ii) might be more effective than (i) at reducing the variance of the estimated PSD while still keeping the bias low. In the sequel, we follow approach (ii).

Let $\beta \geq 1/N$ be the *prespecified (desired) resolution*, and let K be defined by equation (5.3.12): $K = \beta N$. According to the time-bandwidth product result, a bandpass filter with a length-N impulse response may be expected to have a bandwidth on the order of $1/N$ (but not less). Hence, we can cover the preimposed passband

$$[\tilde{\omega} - \beta\pi, \tilde{\omega} + \beta\pi] \tag{5.5.2}$$

(here, $\tilde{\omega}$ stands for the current frequency value) by using $2\pi\beta/(2\pi/N) = K$ filters, which pass essentially nonoverlapping $1/N$-length frequency bands in the interval (5.5.2). *The requirement that the filters' passbands be (nearly) nonoverlapping is a key condition for variance reduction.* In order to see this, let x_p denote the sample obtained at the output of the pth filter:

$$x_p = \sum_{k=0}^{N-1} h_{p,k} y(N-k) = \sum_{t=1}^{N} h_{p,N-t} y(t) \tag{5.5.3}$$

Here, $\{h_{p,k}\}_{k=0}^{N-1}$ is the pth filter's impulse response. The associated frequency response is denoted by $H_p(\omega)$. Note that, in the present case, we consider bandpass filters operating on the raw data in lieu of baseband filters operating on demodulated data (as in RFB). Assume that the *center-frequency gain* of each filter is normalized so that

$$H_p(\tilde{\omega}) = 1, \qquad p = 1, \ldots, K \tag{5.5.4}$$

Then we can write

$$\begin{aligned} E\left\{|x_p|^2\right\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_p(\omega)|^2 \phi(\omega) d\omega \\ &\simeq \frac{1}{2\pi}\int_{\tilde{\omega}-\pi/N}^{\tilde{\omega}+\pi/N} \phi(\omega) d\omega \simeq \frac{2\pi/N}{2\pi}\phi(\tilde{\omega}) = \frac{1}{N}\phi(\tilde{\omega}) \end{aligned} \tag{5.5.5}$$

The second "equality" in (5.5.5) follows from (5.5.4) and the *assumed bandpass characteristics of* $H_p(\omega)$, and the third equality results from the assumption that $\phi(\omega)$ *is approximately constant over the passband*. (Note that the angular frequency passband of $H_p(\omega)$ is $2\pi/N$, as explained before.) In view of (5.5.5), we can estimate $\phi(\tilde{\omega})$ by averaging over the squared magnitudes of the filtered samples $\{x_p\}_{p=1}^{K}$. By doing so, we can achieve a reduction in variance by a factor K, *provided* that the $\{x_p\}$ are statistically independent. (See Section 5.3 for details.) Under the assumption that the filters $\{H_p(\omega)\}$ pass essentially nonoverlapping frequency bands, we readily get (compare (5.3.27))

$$E\left\{x_p x_k^*\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_p(\omega) H_k^*(\omega) \phi(\omega) d\omega \simeq 0 \tag{5.5.6}$$

which implies that the random variables $\{|x_p|^2\}$ are independent, at least under the Gaussian hypothesis. Without the previous assumption on $\{H_p(\omega)\}$, the filtered samples $\{x_p\}$ could be strongly correlated and, therefore, a reduction in variance by a factor K cannot be guaranteed.

The conclusion from the previous (more or less heuristic) discussion is summarized in the following:

> If the passbands of the filters used to cover the prespecified interval (5.5.2) do not overlap, then, by using all filters' output samples—as contrasted to using the output sample of only one filter—we achieve a reduction in the variance of the estimated PSD by a factor equal to the number of filters. The maximum number of such filters that can be found is given by $K = \beta N$. (5.5.7)

By using the insights provided by the previous discussion, as summarized in (5.5.7), we can now approach the bandpass filter's design problem. We sample the frequency axis as in the FFT (as almost any practical implementation of a spectral estimation method does):

$$\tilde{\omega}_s = \frac{2\pi}{N} s \qquad s = 0, \ldots, N-1 \tag{5.5.8}$$

The frequency samples that fall within the passband (5.5.2) are readily seen to be the following:

$$\frac{2\pi}{N}(s+p) \qquad p = -K/2, \ldots, 0, \ldots, K/2 - 1 \tag{5.5.9}$$

(To simplify the discussion, we assume that K is an even integer.) Let

$$H = [h_1 \ldots h_K] \qquad (N \times K) \tag{5.5.10}$$

denote the matrix whose pth column is equal to the impulse-response vector corresponding to the pth bandpass filter. We assume that *the input to the filters is white noise (as in RFB) and design the filters so as to minimize the output power under the constraint that each filter pass undistorted one (and only one) of the frequencies in (5.5.9) (as in Capon).* These design objectives lead to the optimization problem

$$\boxed{\begin{gathered} \min_H (H^*H) \text{ subject to } H^*A = I \\ \text{where } A = \left[a\left(\tfrac{2\pi}{N}\left(s - \tfrac{K}{2}\right)\right), \ldots, a\left(\tfrac{2\pi}{N}\left(s + \tfrac{K}{2} - 1\right)\right)\right] \end{gathered}} \tag{5.5.11}$$

and where $a(\omega) = [1 \ e^{-i\omega} \ldots e^{-i(N-1)\omega}]^T$. Note that the constraint in (5.5.11) guarantees that each frequency in the passband (5.5.9) is passed undistorted by one filter in the set and is annihilated by all the other $(K-1)$ filters. In particular, observe that (5.5.11) implies (5.5.4).

The solution to (5.5.11) follows at once from the result (5.5.1): the minimizing H matrix is given by

$$H = A(A^*A)^{-1} \tag{5.5.12}$$

However, the columns in A are orthogonal

$$A^*A = NI$$

(see (4.3.15)); therefore, (5.5.12) simplifies to

$$\boxed{H = \frac{1}{N}A} \tag{5.5.13}$$

which is the solution of the filter design problem previously formulated.

By using (5.5.13) in (5.5.3), we get

$$\begin{aligned} |x_p|^2 &= \frac{1}{N^2}\left|\sum_{t=1}^{N} e^{i(N-t)\frac{2\pi}{N}(s+p)} y(t)\right|^2 \\ &= \frac{1}{N^2}\left|\sum_{t=1}^{N} y(t)e^{-i\frac{2\pi}{N}(s+p)t}\right|^2 \\ &= \frac{1}{N}\hat{\phi}_p\left(\frac{2\pi}{N}(s+p)\right) \qquad p = -K/2, \ldots, K/2-1 \end{aligned} \tag{5.5.14}$$

where the dependence of $|x_p|^2$ on s (and hence on $\tilde{\omega}_s$) is omitted to simplify the notation and where $\hat{\phi}_p(\omega)$ is the standard periodogram. Finally, (5.5.14), along with (5.5.5), leads to the *FBA spectral estimator*

$$\boxed{\hat{\phi}\left(\frac{2\pi}{N}s\right) = \frac{1}{K}\sum_{p=-K/2}^{K/2-1} N|x_p|^2 = \frac{1}{K}\sum_{l=s-K/2}^{s+K/2-1} \hat{\phi}_p\left(\frac{2\pi}{N}l\right)} \tag{5.5.15}$$

which coincides with the Daniell periodogram estimator (2.7.16). Furthermore, *for* $K = 1$ (i.e., $\beta = 1/N$, which is the choice suitable for "high-resolution" applications), *(5.5.15) reduces to the unmodified periodogram*. Recall also that the RFB method in Section 5.3, for large data lengths, is expected to have a performance similar to that of the Daniell method for $K > 1$ and to that of the basic periodogram for $K = 1$. Hence, in the family of nonparametric spectral estimation methods, the periodograms "are doing well."

5.6 COMPLEMENTS

5.6.1 Another Relationship between the Capon and AR Methods

The relationship between the AR and Capon spectra established in Section 5.4.2 involves all AR spectral models of orders 0 through m. Another interesting relationship, which involves the AR spectrum of order m alone, is presented in this complement.

Let $\hat{\theta} = [\hat{a}_0 \; \hat{a}_1 \ldots \hat{a}_m]^T$ (with $\hat{a}_0 = 1$) denote the vector of the coefficients of the mth-order AR model fitted to the data-sample covariances, and let $\hat{\sigma}^2$ denote the corresponding residual variance. (See Chapter 3 and (5.4.27).) Then the mth-order AR spectrum is given by

$$\hat{\phi}_{AR}(\omega) = \frac{\hat{\sigma}^2}{|a^*(\omega)\hat{\theta}^c|^2} = \frac{\hat{\sigma}^2}{|\sum_{k=0}^{m} \; \hat{a}_k e^{-i\omega k}|^2} \tag{5.6.1}$$

By a simple calculation, $\hat{\phi}_{AR}(\omega)$ can be rewritten as

$$\boxed{\hat{\phi}_{AR}(\omega) = \frac{\hat{\sigma}^2}{\sum_{s=-m}^{m} \; \hat{\rho}(s) e^{i\omega s}}} \tag{5.6.2}$$

where

$$\boxed{\hat{\rho}(s) = \sum_{k=0}^{m-s} \hat{a}_k \hat{a}_{k+s}^* = \hat{\rho}^*(-s), \qquad s = 0, \ldots, m.} \tag{5.6.3}$$

To show this, note that

$$\begin{aligned}
\left| \sum_{k=0}^{m} \hat{a}_k e^{-i\omega k} \right|^2 &= \sum_{k=0}^{m} \sum_{p=0}^{m} \hat{a}_k \hat{a}_p^* e^{-i\omega(k-p)} = \sum_{k=0}^{m} \sum_{s=k-m}^{k} \hat{a}_k \hat{a}_{k-s}^* e^{-i\omega s} \\
&= \sum_{k=0}^{m} \sum_{s=-m}^{m} \hat{a}_k \hat{a}_{k-s}^* e^{-i\omega s} = \sum_{s=-m}^{m} \sum_{k=0}^{m} \hat{a}_k \hat{a}_{k+s}^* e^{i\omega s} \\
&= \sum_{s=-m}^{m} \left(\sum_{k=0}^{m-s} \hat{a}_k \hat{a}_{k+s}^* \right) e^{i\omega s}
\end{aligned}$$

and (5.6.2)–(5.6.3) immediately follow.

Next, assume that the (sample) covariance matrix $\hat{R}$ is Toeplitz. (We note, in passing, that this is a minor restriction for the temporal spectral estimation problem of this chapter, but it can be quite a restrictive assumption for the spatial problem of the next chapter.) Then the Capon spectrum in equation (5.4.19) (with the factor $m+1$ omitted, for convenience) can be written as

$$\boxed{\hat{\phi}_{CM}(\omega) = \frac{\hat{\sigma}^2}{\sum_{s=-m}^{m} \hat{\mu}(s) e^{i\omega s}}} \tag{5.6.4}$$

where

$$\boxed{\hat{\mu}(s) = \sum_{k=0}^{m-s} (m+1-2k-s)\hat{a}_k \hat{a}_{k+s}^* = \hat{\mu}^*(-s), \qquad s = 0, \ldots, m} \tag{5.6.5}$$

To prove (5.6.4), we make use of the Gohberg–Semencul (GS) formula derived in Complement 3.9.4, which is repeated here for convenience:

$$\hat{\sigma}^2 \hat{R}^{-1} = \begin{bmatrix} 1 & \cdots & \cdots & 0 \\ \hat{a}_1^* & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \hat{a}_m^* & \cdots & \hat{a}_1^* & 1 \end{bmatrix} \begin{bmatrix} 1 & \hat{a}_1 & \cdots & \hat{a}_m \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \hat{a}_1 \\ 0 & \cdots & \cdots & 1 \end{bmatrix}$$

$$- \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \hat{a}_m & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \hat{a}_1 & \cdots & \hat{a}_m & 0 \end{bmatrix} \begin{bmatrix} 0 & \hat{a}_m^* & \cdots & \hat{a}_1^* \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \hat{a}_m^* \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

This formula is, in fact, the complex conjugate of the GS formula in Complement 3.9.4, because the preceding matrix $\hat{R}$ is the complex conjugate of the one considered in Chapter 3.

For the sake of convenience, let $\hat{a}_k = 0$ for $k \notin [0,\ m]$. By making use of this convention and of the GS formula, we obtain

$$\begin{aligned} f(\omega) &\triangleq \hat{\sigma}^2 a^*(\omega) \hat{R}^{-1} a(\omega) \\ &= \sum_{p=0}^{m} \left\{ \left| \sum_{k=0}^{m} \hat{a}_{k-p}\ e^{-i\omega k} \right|^2 - \left| \sum_{k=0}^{m} \hat{a}_{m+1-k+p}^* e^{-i\omega k} \right|^2 \right\} \\ &= \sum_{p=0}^{m} \sum_{k=0}^{m} \sum_{\ell=0}^{m} (\hat{a}_{k-p} \hat{a}_{\ell-p}^* - \hat{a}_{m+1+p-k}^* \hat{a}_{m+1-\ell+p}) e^{i\omega(\ell-k)} \\ &= \sum_{\ell=0}^{m} \sum_{p=0}^{m} \sum_{s=\ell-m}^{\ell} (\hat{a}_{\ell-s-p} \hat{a}_{\ell-p}^* - \hat{a}_{m+1-\ell+s+p}^* \hat{a}_{m+1+p-\ell}) e^{i\omega s} \end{aligned} \tag{5.6.6}$$

where the last equality has been obtained by the substitution $s = \ell - k$. Next, make the substitution $j = \ell - p$ in (5.6.6) to get

$$f(\omega) = \sum_{\ell=0}^{m} \sum_{j=\ell-m}^{\ell} \sum_{s=\ell-m}^{\ell} (\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*)e^{i\omega s} \tag{5.6.7}$$

Since $\hat{a}_{j-s} = 0$ and $\hat{a}_{m+1+s-j}^* = 0$ for $s > j$, we can extend the summation over s in (5.6.7) up to $s = m$. Furthermore, the summand in (5.6.7) is zero for $j < 0$; hence, we can truncate the summation over j to the interval $[0,\ \ell]$. These two observations yield

$$f(\omega) = \sum_{\ell=0}^{m} \sum_{j=0}^{\ell} \sum_{s=\ell-m}^{m} (\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*)e^{i\omega s} \tag{5.6.8}$$

Next, decompose $f(\omega)$ additively as

$$f(\omega) = T_1(\omega) + T_2(\omega)$$

where

$$T_1(\omega) = \sum_{\ell=0}^{m} \sum_{j=0}^{\ell} \sum_{s=0}^{m} (\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*)e^{i\omega s}$$

$$T_2(\omega) = \sum_{\ell=0}^{m} \sum_{j=0}^{\ell} \sum_{s=\ell-m}^{-1} (\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*)e^{i\omega s}$$

(The term in T_2 corresponding to $\ell = m$ is zero.) Let

$$\hat{\mu}(s) \triangleq \sum_{\ell=0}^{m} \sum_{j=0}^{\ell} (\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*) \tag{5.6.9}$$

By using this notation, we can write $T_1(\omega)$ as

$$T_1(\omega) = \sum_{s=0}^{m} \hat{\mu}(s)e^{i\omega s}$$

Since $f(\omega)$ is real valued for any $\omega \in [-\pi,\ \pi]$, we must also have

$$T_2(\omega) = \sum_{s=-1}^{-m} \hat{\mu}^*(-s)e^{i\omega s}$$

As the summand in (5.6.9) does not depend on ℓ, we readily obtain

$$\hat{\mu}(s) = \sum_{j=0}^{m} (m+1-j)\,(\hat{a}_{j-s}\hat{a}_j^* - \hat{a}_{m+1-j}\hat{a}_{m+1+s-j}^*)$$

$$= \sum_{k=0}^{m-s} (m+1-k-s)\,\hat{a}_k\hat{a}_{k+s}^* - \sum_{k=1}^{m} k\hat{a}_k\hat{a}_{k+s}^*$$

$$= \sum_{k=0}^{m-s} (m+1-2k-s)\hat{a}_k\hat{a}_{k+s}^*$$

which coincides with (5.6.5). Thus, the proof of (5.6.4) is concluded.

Remark: The reader might wonder what happens with the formulas just derived if the AR model parameters are calculated by using the same sample covariance matrix as in the Capon estimator. In such a case, the parameters $\{\hat{a}_k\}$ in (5.6.1) and in the preceding GS formula should be replaced by $\{\hat{a}_k^*\}$ (see (5.4.27)). Consequently, both (5.6.2)–(5.6.3) and (5.6.4)–(5.6.5) continue to hold, but with $\{\hat{a}_k\}$ replaced by $\{\hat{a}_k^*\}$ (and $\{\hat{a}_k^*\}$ replaced by $\{\hat{a}_k\}$). ■

By comparing (5.6.2) and (5.6.4), we see that the reciprocals of both $\hat{\phi}_{AR}(\omega)$ and $\hat{\phi}_{CM}(\omega)$ have the form of a Blackman–Tukey spectral estimate associated with the "covariance sequences" $\{\hat{\rho}(s)\}$ and $\{\hat{\mu}(s)\}$, respectively. The only difference between $\hat{\phi}_{AR}(\omega)$ and $\hat{\phi}_{CM}(\omega)$ is that the sequence $\{\hat{\mu}(s)\}$ corresponding to $\hat{\phi}_{CM}(\omega)$ is a "linearly tapered" version of the sequence $\{\hat{\rho}(s)\}$ corresponding to $\hat{\phi}_{AR}(\omega)$. Much as with the interpretation in Section 5.4.2, the previous observation can be used to understand intuitively why the Capon spectral estimates are smoother and have poorer resolution than the AR estimates of the same order. (For more details on this aspect and other aspects related to the discussion in this complement, see [Musicus 1985].)

We remark in passing that the name "covariance sequence" given, for example, to $\{\hat{\rho}(s)\}$, is not coincidental: The $\{\hat{\rho}(s)\}$ are so-called *sample inverse covariances* associated with $\hat{R}$, and they can be shown to possess a number of interesting and useful properties. (See, for example, [Cleveland 1972; Bhansali 1980].)

The formula (5.6.4) can be used for the computation of $\hat{\phi}_{CM}(\omega)$, as we now show. Assuming that $\hat{R}$ is already available, we can use the Levinson–Durbin algorithm to compute $\{\hat{a}_k\}$ and $\hat{\sigma}^2$ (and thus $\{\hat{\mu}(s)\}$) in $\mathcal{O}(m^2)$ flops. Then (5.6.4) can be evaluated at M Fourier frequencies (say) by using the FFT. The resulting total computational burden is on the order of $\mathcal{O}(m^2 + M\log_2 M)$ flops. For commonly encountered values of m and M, this is about m times smaller than the burden associated with the eigendecomposition-based computational procedure of Exercise 5.5. Note, however, that the latter algorithm can be applied to a general $\hat{R}$ matrix, whereas the one derived in this complement is limited to Toeplitz covariance matrices. Finally, note that the extension of the results in this complement to two-dimensional (2D) signals can be found in [Jakobsson, Marple, and Stoica 2000].

5.6.2 Multiwindow Interpretation of Daniell and Blackman–Tukey Periodograms

As stated in Exercise 5.1, the Bartlett and Welch periodograms can be cast into the multiwindow framework of Section 5.3.3. In other words, they can be written in the following form (see (5.7.1)):

$$\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} w_{p,t}\, y(t) e^{-i\omega t} \right|^2 \tag{5.6.10}$$

for certain temporal (or data) windows $\{w_{p,t}\}$ (also called *tapers*). Here, K denotes the number of windows used by the method in question.

In this complement, we show that the Daniell periodogram and the Blackman–Tukey periodogram (with some commonly used lag windows) can also be interpreted as multiwindow methods. Unlike the approximate multiwindow interpretation of a spectrally smoothed periodogram described in Section 5.3.3 (see equations (5.3.31)–(5.3.33) there), the multiwindow interpretations presented in this complement are *exact*. More details on the topic of this complement can be found in [McCloud, Scharf, and Mullis 1999], where it is also shown that the Blackman–Tukey periodogram with any "good" window can be cast in a multiwindow framework, but only approximately.

We begin by writing (5.6.10) as a quadratic form in the data sequence. Let

$$z(\omega) = \begin{bmatrix} y(1)e^{-i\omega} \\ \vdots \\ y(N)e^{-iN\omega} \end{bmatrix}, \qquad (N \times 1)$$

$$W = \begin{bmatrix} w_{1,1} & \cdots & w_{1,N} \\ \vdots & & \vdots \\ w_{K,1} & \cdots & w_{K,N} \end{bmatrix}, \qquad (K \times N)$$

and let $[x]_p$ denote the pth element of a vector x. Using this notation, we can rewrite (5.6.10) in the desired form

$$\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| [Wz(\omega)]_p \right|^2$$

or

$$\boxed{\hat{\phi}(\omega) = \frac{1}{K} z^*(\omega) W^* W z(\omega)} \tag{5.6.11}$$

which is a quadratic form in $z(\omega)$. The rank of the matrix W^*W is less than or equal to K; typically, $\text{rank}(W^*W) = K \ll N$.

Next, we turn our attention to the Daniell periodogram (see (2.7.16))

$$\hat{\phi}_D(\omega) = \frac{1}{2J+1} \sum_{j=-J}^{J} \hat{\phi}_p \left(\omega + j \frac{2\pi}{N} \right) \tag{5.6.12}$$

where $\hat{\phi}_p(\omega)$ is the standard periodogram given in (2.2.1):

$$\hat{\phi}_p(\omega) = \frac{1}{N} \left| \sum_{t=1}^{N} y(t) e^{-i\omega t} \right|^2$$

Letting

$$a_j^* = \left[e^{-i\frac{2\pi}{N}j}, e^{-i\frac{2\pi}{N}(2j)}, \ldots, e^{-i\frac{2\pi}{N}(Nj)} \right] \tag{5.6.13}$$

we can write

$$\begin{aligned} \hat{\phi}_p \left(\omega + j \frac{2\pi}{N} \right) &= \frac{1}{N} \left| \sum_{t=1}^{N} y(t) e^{-i\omega t} e^{-i\frac{2\pi}{N}(jt)} \right|^2 \\ &= \frac{1}{N} \left| a_j^* z(\omega) \right|^2 = \frac{1}{N} z^*(\omega) a_j a_j^* z(\omega) \end{aligned} \tag{5.6.14}$$

which implies that

$$\boxed{\hat{\phi}_D(\omega) = \frac{1}{N(2J+1)} z^*(\omega) W_D^* W_D z(\omega)} \tag{5.6.15}$$

where

$$\boxed{W_D = \left[a_{-J}, \ldots, a_0, \ldots, a_J \right]^*, \qquad (2J+1) \times N} \tag{5.6.16}$$

This establishes the fact that *the Daniell periodogram can be interpreted as a multiwindow method by using $K = 2J+1$ tapers, given by* (5.6.16). Like the tapers used by the seemingly more elaborate RFB approach, the Daniell periodogram tapers can also be motivated by using a sound design methodology. (See Section 5.5.)

In the remaining part of this complement, we consider the Blackman–Tukey periodogram in (2.5.1) with a window of length $M = N$:

$$\hat{\phi}_{BT}(\omega) = \sum_{k=-(N-1)}^{N-1} w(k) \hat{r}(k) e^{-i\omega k} \tag{5.6.17}$$

A commonly used class of windows, including the Hanning and Hamming windows in Table 2.1, is described by the equation

$$w(k) = \alpha + \beta \cos(\Delta k) = \left(\alpha + \frac{\beta}{2} e^{i\Delta k} + \frac{\beta}{2} e^{-i\Delta k}\right) \tag{5.6.18}$$

for various parameters α, β, and Δ. Inserting (5.6.18) into (5.6.17) yields

$$\begin{aligned} \hat{\phi}_{BT}(\omega) &= \sum_{k=-(N-1)}^{N-1} \left(\alpha + \frac{\beta}{2} e^{i\Delta k} + \frac{\beta}{2} e^{-i\Delta k}\right) \hat{r}(k) e^{-i\omega k} \\ &= \alpha \hat{\phi}_p(\omega) + \frac{\beta}{2} \hat{\phi}_p(\omega - \Delta) + \frac{\beta}{2} \hat{\phi}_p(\omega + \Delta) \end{aligned} \tag{5.6.19}$$

where $\hat{\phi}_p(\omega)$ is the standard periodogram given by (2.2.1) or, equivalently, by (2.2.2):

$$\hat{\phi}_p(\omega) = \sum_{k=-(N-1)}^{N-1} \hat{r}(k) e^{-i\omega k}$$

Comparing (5.6.19) with (5.6.12) (and with (5.6.14)–(5.6.16)) allows us to write

$$\boxed{\hat{\phi}_{BT}(\omega) = \frac{1}{N} z^*(\omega) W_{BT}^* W_{BT} z(\omega)} \tag{5.6.20}$$

where

$$\boxed{W_{BT} = \left[\sqrt{\frac{\beta}{2}} a_{-\Delta}, \sqrt{\alpha} a_0, \sqrt{\frac{\beta}{2}} a_{\Delta}\right]^*, \qquad (3 \times N)} \tag{5.6.21}$$

for $\alpha, \beta \geq 0$ and where a_Δ is given by (similarly to a_j in (5.6.13))

$$a_\Delta^* = \left[e^{-i\Delta}, \ldots, e^{-i\Delta N}\right]$$

Hence, we conclude that *the Blackman–Tukey periodogram with a Hamming or Hanning window (or any other window having the form of (5.6.18)) can be interpreted as a multiwindow method by using $K = 3$ tapers given by (5.6.21).* Similarly, $\hat{\phi}_{BT}(\omega)$ using the Blackman window in Table 2.1 can be shown to be equivalent to a multiwindow method with $K = 7$ tapers.

Interestingly, as a by-product of the analysis in this complement, we note from (5.6.19) that the Blackman–Tukey periodogram with a window of the form in (5.6.18) can be *very efficiently* computed from the values of the standard periodogram. Because the Blackman window has a form similar to (5.6.18), $\hat{\phi}_{BT}(\omega)$ using the Blackman window can similarly be implemented in an efficient way. This way of computing $\hat{\phi}_{BT}(\omega)$ is faster than the method outlined in Complement 2.8.2 for a general lag window.

5.6.3 Capon Method for Exponentially Damped Sinusoidal Signals

The signals that are dealt with in some applications of spectral analysis, such as in magnetic resonance spectroscopy, consist of a sum of *exponentially damped sinusoidal components* (damped sinusoids, for short), instead of the pure sinusoids as in (4.1.1). Such signals are described by the equation

$$y(t) = \sum_{k=1}^{n} \beta_k e^{(\rho_k + i\omega_k)t} + e(t), \qquad t = 1, \ldots, N \tag{5.6.22}$$

where β_k and ω_k are the amplitude and frequency of the kth component (as in Chapter 4), and $\rho_k < 0$ is the so-called damping parameter. The (noise-free) signal in (5.6.22) is *nonstationary*; hence, it does not have a power spectral density. However, it possesses an *amplitude spectrum* that is defined as follows:

$$|\beta(\rho, \omega)| = \begin{cases} |\beta_k|, & \text{for } \omega = \omega_k, \rho = \rho_k \quad (k = 1, \ldots, n) \\ 0, & \text{elsewhere} \end{cases} \tag{5.6.23}$$

Furthermore, because an exponentially damped sinusoid satisfies the finite-energy condition in (1.2.1), the (noise-free) signal in (5.6.22) also possesses an *energy spectrum*. As with (5.6.23), we can define the energy spectrum of the damped sinusoidal signal in (5.6.22) as a 2D function of (ρ, ω) that consists of n pulses at $\{\rho_k, \omega_k\}$, where the height of the function at each of these points is equal to the energy of the corresponding component. The energy of a generic component with parameters (β, ρ, ω) is given by

$$\sum_{t=1}^{N} \left|\beta e^{(\rho + i\omega)t}\right|^2 = |\beta|^2 e^{2\rho} \sum_{t=0}^{N-1} e^{2\rho t} = |\beta|^2 e^{2\rho} \frac{1 - e^{2\rho N}}{1 - e^{2\rho}} \tag{5.6.24}$$

It follows from (5.6.24) and the foregoing discussion that the energy spectrum can be expressed as a function of the amplitude spectrum in (5.6.23), via the formula

$$E(\rho, \omega) = |\beta(\rho, \omega)|^2 L(\rho) \tag{5.6.25}$$

where

$$L(\rho) = e^{2\rho} \frac{1 - e^{2\rho N}}{1 - e^{2\rho}} \tag{5.6.26}$$

The amplitude spectrum, and hence the energy spectrum, of the signal in (5.6.22) can be estimated by using *an extension of the Capon method* that is introduced in Section 5.4. To develop this extension, we consider the data vector:

$$\tilde{y}(t) = \left[y(t), y(t+1), \ldots, y(t+m)\right]^T \tag{5.6.27}$$

(Alternatively, we could have used the data vector defined in (5.4.2).) Let h denote the coefficient vector of the Capon FIR filter as in (5.4.1). Then, the output of the filter with the data vector in

(5.6.27) is given by

$$\tilde{y}_F(t) = h^*\tilde{y}(t) = h^*\begin{bmatrix} y(t) \\ \vdots \\ y(t+m) \end{bmatrix}, \qquad t = 1, \dots, N-m \tag{5.6.28}$$

To derive Capon-like estimates of the amplitude and energy spectra of (5.6.22), let

$$\hat{R} = \frac{1}{N-m}\sum_{t=1}^{N-m} \tilde{y}(t)\tilde{y}^*(t) \tag{5.6.29}$$

denote the sample covariance matrix of the data vector in (5.6.27). Then the sample variance of the filter output can be written as

$$\frac{1}{N-m}\sum_{t=1}^{N-m} |\tilde{y}_F(t)|^2 = h^*\hat{R}h \tag{5.6.30}$$

By definition, the Capon filter minimizes (5.6.30) under the constraint that the filter pass, without distortion, a generic damped sinusoid with parameters (β, ρ, ω). The filter output corresponding to such a generic component is given by

$$h^*\begin{bmatrix} \beta e^{(\rho+i\omega)t} \\ \beta e^{(\rho+i\omega)(t+1)} \\ \vdots \\ \beta e^{(\rho+i\omega)(t+m)} \end{bmatrix} = \left(h^*\begin{bmatrix} 1 \\ e^{\rho+i\omega} \\ \vdots \\ e^{(\rho+i\omega)m} \end{bmatrix} \right) \beta e^{(\rho+i\omega)t} \tag{5.6.31}$$

Hence, the distortionless filtering constraint can be expressed as

$$h^*a(\rho, \omega) = 1 \tag{5.6.32}$$

where

$$\boxed{a(\rho, \omega) = \left[1, e^{\rho+i\omega}, \dots, e^{(\rho+i\omega)m}\right]^T} \tag{5.6.33}$$

The minimizer of the quadratic function in (5.6.30) under the linear constraint (5.6.32) is given by the familiar formula (see (5.4.7)–(5.4.8))

$$\boxed{h(\rho, \omega) = \frac{\hat{R}^{-1}a(\rho, \omega)}{a^*(\rho, \omega)\hat{R}^{-1}a(\rho, \omega)}} \tag{5.6.34}$$

where we have stressed, via notation, the dependence of h on both ρ and ω.

The output of the filter in (5.6.34), due to a possible (generic) damped sinusoid in the signal with parameters (β, ρ, ω), is given (*cf.* (5.6.31) or (5.6.32)) by

$$h^*(\rho, \omega)\tilde{y}(t) = \beta e^{(\rho+i\omega)t} + e_F(t), \qquad t = 1, \ldots, N-m \tag{5.6.35}$$

where $e_F(t)$ denotes the filter output due to noise and to any other signal components. For given (ρ, ω), the least-squares estimate of β in (5.6.35) (see, e.g., Result R32 in Appendix A) is

$$\hat{\beta}(\rho, \omega) = \frac{\sum_{t=1}^{N-m} h^*(\rho, \omega)\tilde{y}(t)e^{(\rho-i\omega)t}}{\sum_{t=1}^{N-m} e^{2\rho t}} \tag{5.6.36}$$

Let $\tilde{L}(\rho)$ be defined similarly to $L(\rho)$ in (5.6.26), but with N replaced by $N-m$, and let

$$\boxed{\tilde{Y}(\rho, \omega) = \frac{1}{\tilde{L}(\rho)} \sum_{t=1}^{N-m} \tilde{y}(t)e^{(\rho-i\omega)t}} \tag{5.6.37}$$

It follows from (5.6.36), along with (5.6.25), that Capon-like estimates of the amplitude spectrum and energy spectrum of the signal in (5.6.22) can be obtained, respectively, as

$$\boxed{\left|\hat{\beta}(\rho, \omega)\right| = \left|h^*(\rho, \omega)\tilde{Y}(\rho, \omega)\right|} \tag{5.6.38}$$

and as

$$\boxed{\hat{E}(\rho, \omega) = \left|\hat{\beta}(\rho, \omega)\right|^2 L(\rho)} \tag{5.6.39}$$

Remark: We could have estimated the amplitude, β, of a generic component with parameters (β, ρ, ω) directly from the unfiltered data samples $\{y(t)\}_{t=1}^{N}$. However, the use of the Capon-filtered data in (5.6.35) usually leads to enhanced performance. The main reason for this performance gain lies in the fact that the SNR corresponding to the generic component in the filtered data is typically much higher than that in the raw data, owing to the good rejection properties of the Capon filter. This higher SNR leads to more accurate amplitude estimates, in spite of the loss of m data samples in the filtering operation in (5.6.35). ■

Finally, we note that the sample Capon energy or amplitude spectrum can be used *to estimate the signal parameters* $\{\beta_k, \rho_k, \omega_k\}$ in a standard manner. Specifically, we compute either $|\hat{\beta}(\rho, \omega)|$ or $\hat{E}(\rho, \omega)$ at the points of a fine grid covering the region of interest in the two-dimensional (ρ, ω)

plane and obtain estimates of (ρ_k, ω_k) as the locations of the n largest spectral peaks; estimates of β_k can then be derived from (5.6.36), with (ρ, ω) replaced by the estimated values of (ρ_k, ω_k). There is empirical evidence that the use of $\hat{E}(\rho, \omega)$ in general leads to (slightly) more accurate signal parameter estimates than does the use of $|\hat{\beta}(\rho, \omega)|$; see [Stoica and Sundin 2001]. For more details on the topic of this complement, including the computation of the two-dimensional spectra in (5.6.38) and (5.6.39), we refer the reader to [Stoica and Sundin 2001].

5.6.4 Amplitude and Phase Estimation Method (APES)

The design idea behind the Capon filter is based on the following two principles, as discussed in Section 5.4:

(a) the sinusoid with frequency ω (currently considered in the analysis) passes through the filter in as distortionless a manner as possible; and
(b) any other frequencies in the data (corresponding, e.g., to other sinusoidal components in the signal or to noise) are suppressed by the filter as much as possible.

The output of the filter whose input is a sinusoid with frequency ω, $\{\beta e^{i\omega t}\}$, is given (assuming forward filtering, as in (5.4.2)) by

$$h^* \begin{bmatrix} e^{i\omega t} \\ e^{i\omega(t-1)} \\ \vdots \\ e^{i\omega(t-m)} \end{bmatrix} \beta = \left(h^* \begin{bmatrix} 1 \\ e^{-i\omega} \\ \vdots \\ e^{-i\omega m} \end{bmatrix} \right) \beta e^{i\omega t} \tag{5.6.40}$$

For backward filtering, as used in Complement 5.6.3, a similar result can be derived. It follows from (5.6.40) that the design objective in (a) can be expressed mathematically via the linear constraint on h

$$h^* a(\omega) = 1 \tag{5.6.41}$$

where

$$a(\omega) = \left[1, e^{-i\omega}, \ldots, e^{-i\omega m}\right]^T \tag{5.6.42}$$

(See (5.4.5)–(5.4.7).) Regarding the second design objective, its statement in (b) is sufficiently general to allow several different mathematical formulations. The Capon method is based on the idea that the goal in (b) is achieved if the power at the filter output is minimized. (See (5.4.7).) In this complement, another way to formulate (b) mathematically is described.

At a given frequency ω, let us choose h such that the filter output, $\{h^* \tilde{y}(t)\}$, where

$$\tilde{y}(t) = \left[y(t), y(t-1), \ldots, y(t-m)\right]^T$$

is as close as possible in a least-squares (LS) sense to a sinusoid with frequency ω and constant amplitude β. Mathematically, we obtain both h and β, for a given ω, by minimizing the LS criterion

$$\boxed{\min_{h,\beta} \frac{1}{N-m} \sum_{t=m+1}^{N} \left| h^* \tilde{y}(t) - \beta e^{i\omega t} \right|^2 \quad \text{subject to } h^* a(\omega) = 1} \tag{5.6.43}$$

Note that the estimation of the amplitude and phase (i.e., $|\beta|$ and $\arg(\beta)$) of the sinusoid with frequency ω is an intrinsic part of the method based on (5.6.43). This observation motivates the name *Amplitude and Phase EStimation* (APES) given to the method described by (5.6.43).

Because (5.6.43) is a linearly constrained quadratic problem, we should be able to find its solution in closed form. Let

$$\boxed{g(\omega) = \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t) e^{-i\omega t}} \tag{5.6.44}$$

Then a straightforward calculation shows that the criterion function in (5.6.43) can be rewritten as

$$\begin{aligned}
&\frac{1}{N-m} \sum_{t=m+1}^{N} \left| h^* \tilde{y}(t) - \beta e^{i\omega t} \right|^2 \\
&\quad = h^* \hat{R} h - \beta^* h^* g(\omega) - \beta g^*(\omega) h + |\beta|^2 \\
&\quad = \left| \beta - h^* g(\omega) \right|^2 + h^* \hat{R} h - \left| h^* g(\omega) \right|^2 \\
&\quad = \left| \beta - h^* g(\omega) \right|^2 + h^* \left[\hat{R} - g(\omega) g^*(\omega) \right] h
\end{aligned} \tag{5.6.45}$$

where

$$\hat{R} = \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t) \tilde{y}^*(t) \tag{5.6.46}$$

(See (5.4.18).) The minimization of (5.6.45) with respect to β is immediate:

$$\beta(\omega) = h^* g(\omega) \tag{5.6.47}$$

Inserting (5.6.47) into (5.6.45) yields the following problem, whose solution will determine the filter coefficient vector:

$$\min_{h} h^* \hat{Q}(\omega) h \qquad \text{subject to } h^* a(\omega) = 1 \tag{5.6.48}$$

In this equation

$$\hat{Q}(\omega) = \hat{R} - g(\omega) g^*(\omega) \tag{5.6.49}$$

As (5.6.48) has the same form as the Capon filter-design problem (see (5.4.7)), the solution to (5.6.48) is readily derived (compare (5.4.8)):

$$h(\omega) = \frac{\hat{Q}^{-1}(\omega)a(\omega)}{a^*(\omega)\hat{Q}^{-1}(\omega)a(\omega)} \tag{5.6.50}$$

A direct implementation of (5.6.50) would require the inversion of the matrix $\hat{Q}(\omega)$ for each value of $\omega \in [0, 2\pi]$ considered. To avoid such an intensive computational task, we can use the matrix inversion lemma (Result R27 in Appendix A) to express the inverse in (5.6.50) as follows:

$$\hat{Q}^{-1}(\omega) = \left[\hat{R} - g(\omega)g^*(\omega)\right]^{-1} = \hat{R}^{-1} + \frac{\hat{R}^{-1}g(\omega)g^*(\omega)\hat{R}^{-1}}{1 - g^*(\omega)\hat{R}^{-1}g(\omega)} \tag{5.6.51}$$

Inserting (5.6.51) into (5.6.50) yields the following expression for the *APES filter*:

$$\boxed{h(\omega) = \frac{\left[1 - g^*(\omega)\hat{R}^{-1}g(\omega)\right]\hat{R}^{-1}a(\omega) + \left[g^*(\omega)\hat{R}^{-1}a(\omega)\right]\hat{R}^{-1}g(\omega)}{\left[1 - g^*(\omega)\hat{R}^{-1}g(\omega)\right]a^*(\omega)\hat{R}^{-1}a(\omega) + \left|a^*(\omega)\hat{R}^{-1}g(\omega)\right|^2}} \tag{5.6.52}$$

From (5.6.47) and (5.6.52), we obtain the following formula for *the APES estimate of the (complex) amplitude spectrum* (see Complement 5.6.3 for a definition of the amplitude spectrum):

$$\boxed{\beta(\omega) = \frac{a^*(\omega)\hat{R}^{-1}g(\omega)}{\left[1 - g^*(\omega)\hat{R}^{-1}g(\omega)\right]a^*(\omega)\hat{R}^{-1}a(\omega) + \left|a^*(\omega)\hat{R}^{-1}g(\omega)\right|^2}} \tag{5.6.53}$$

Comparing this with the *Capon estimate of the amplitude spectrum*, given by

$$\beta(\omega) = \frac{a^*(\omega)\hat{R}^{-1}g(\omega)}{a^*(\omega)\hat{R}^{-1}a(\omega)} \tag{5.6.54}$$

we see that the APES estimate in (5.6.53) is more involved computationally, but not by much.

Remark: Our discussion has focused on the estimation of the amplitude spectrum. If the power spectrum is what we want to estimate, then we can use the APES filter, (5.6.52), in the PSD estimation approach described in Section 5.4, or we can simply take $|\beta(\omega)|^2$ (along with a possible scaling) as an estimate of the PSD. ■

This derivation of APES is adapted from [Stoica, Li, and Li 1999]. The original derivation of APES, provided in [Li and Stoica 1996a], was different: It was based on an approximate maximum likelihood approach. We refer the reader to [Li and Stoica 1996a] for the original derivation of APES and for many other details on this approach to spectral analysis.

We end this complement with a brief comparison of Capon and APES from a performance standpoint. Extensive empirical and analytical studies of these two methods (see, e.g., [Larsson, Li, and Stoica 2003] and its references) have shown that Capon has a (slightly) higher resolution than APES and also that the Capon estimates of the frequencies of a multicomponent sinusoidal signal in noise are more accurate than the APES estimates. On the other hand, for a given set of frequency estimates $\{\hat{\omega}_k\}$ in the vicinity of the true frequencies, the APES estimates of the amplitudes $\{\beta_k\}$ are much more accurate than the Capon estimates; the Capon estimates are always biased towards zero, sometimes significantly so. This suggests that, at least for line-spectral analysis, a method better than either Capon or APES can be obtained by combining them in the following way:

- Estimate the frequencies $\{\omega_k\}$ as the locations of the dominant peaks of the Capon spectrum.
- Estimate the amplitudes $\{\beta_k\}$ by using the APES formula (5.6.53) evaluated at the frequency estimates obtained in the previous step.

This *combined Capon-APES (CAPES) method* was introduced in [Jakobsson and Stoica 2000].

5.6.5 Amplitude and Phase Estimation Method for Gapped Data (GAPES)

In some applications of spectral analysis, the data sequence has gaps; these gaps may be due to the failure of a measuring device or due to the impossibility of performing measurements for some periods of time (such as in astronomy). In this complement, we will present an extension of the Amplitude and Phase EStimation (APES) method, outlined in Complement 5.6.4, to *gapped-data sequences*. Gapped-data sequences are evenly sampled data strings that contain unknown samples which are usually, but not always, clustered together in groups of reasonable size. We will use the acronym GAPES to designate the extended approach.

Most of the available methods for the spectral analysis of gapped data perform (either implicitly or explicitly) an interpolation of the missing data, followed by a standard full-data spectral analysis. The data interpolation step is critical, and it cannot be completed without making (sometimes hidden) assumptions on the data sequence. For example, one such assumption is that the data is bandlimited with a known cutoff frequency. Intuitively, these assumptions can be viewed as attempts to add extra "information" to the spectral analysis problem, which might be able to compensate for the information lost due to the missing data samples. The problem with these assumptions, though, is that they are not generally easy to check in applications, either *a priori* or *a posteriori*. The GAPES approach presented here is based on the sole assumption that *the spectral content of the missing data is similar to that of the available data*. This assumption is very natural, and one could argue that it introduces no restriction at all.

We begin the derivation of GAPES by rewriting the APES least-squares fitting criterion (see equation (5.6.43) in Complement 5.6.4) in a form that is more convenient for the discussion here. Specifically, we use the notation $h(\omega)$ and $\beta(\omega)$ to stress the dependence on ω of both the APES filter and the amplitude spectrum. Also, we note that, in applications, the frequency variable is usually sampled as

$$\omega_k = \frac{2\pi}{K}k, \qquad k = 1, \ldots, K \tag{5.6.55}$$

where K is an integer (much) larger than N. Making use of the above notation and (5.6.55), we rewrite the APES criterion as follows:

$$\begin{aligned} &\min \sum_{k=1}^{K} \sum_{t=m+1}^{N} \left| h^*(\omega_k)\tilde{y}(t) - \beta(\omega_k)e^{i\omega_k t} \right|^2 \\ &\text{subject to } h^*(\omega_k)a(\omega_k) = 1 \text{ for } k = 1, \ldots, K \end{aligned} \tag{5.6.56}$$

Evidently, the minimization of the criterion in (5.6.56) with respect to $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$ reduces to the minimization of the inner sum in (5.6.56) for each k. Hence, in the full-data case, the problem in (5.6.56) is equivalent to the standard APES problem in equation (5.6.43) in Complement 5.6.4. However, in the gapped-data case, the form of the APES criterion in (5.6.56) turns out to be more convenient than that in (5.6.43), as we shall subsequently see.

To continue, we need some additional notation. Let

$$y_a = \text{the vector containing the available samples in } \{y(t)\}_{t=1}^{N}$$

$$y_u = \text{the vector containing the unavailable samples in } \{y(t)\}_{t=1}^{N}$$

The main idea behind the GAPES approach is to minimize (5.6.56) with respect to both $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$ *as well as* with respect to y_u. Such a formulation of the gapped-data problem is appealing, because it leads to

(i) an analysis filter bank $\{h(\omega_k)\}$ for which the filtered sequence is as close as possible in a LS sense to the (possible) sinusoidal component in the data that has frequency ω_k, which is the main design goal in the filter-bank approach to spectral analysis; and

(ii) an estimate of the missing samples in y_u whose spectral content mimics the spectral content of the available data as much as possible in the LS sense of (5.6.56).

The criterion in (5.6.56) is a *quartic* function of the unknowns $\{h(\omega_k)\}$, $\{\beta(\omega_k)\}$, and y_u. Consequently, in general, its minimization requires the use of an iterative algorithm; that is, a closed-form solution is unlikely to exist. The GAPES method uses a *cyclic minimizer* to minimize the criterion in (5.6.56). (See Complement 4.9.5 for a general description of cyclic minimizers.) A *step-by-step description of GAPES* is given in the following box.

To reduce the computational burden of the GAPES algorithm, we can run it with a value of K that is not much larger than N (e.g., $K \in [2N, 4N]$). After the iterations are terminated, the final spectral estimate can be evaluated on a (much) finer frequency grid, if desired.

The GAPES Algorithm

Step 0. Obtain initial estimates of $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$.

Step 1. Use the most recent estimates of $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$ to estimate y_u via the minimization of (5.6.56).

Step 2. Use the most recent estimate of y_u to estimate $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$ via the minimization of (5.6.56).

Step 3. Check on the convergence of the iteration—for example, by checking whether the relative change of the criterion between two consecutive iterations is smaller than a preassigned value. If *no*, then go to Step 1. If *yes*, then we have a final amplitude spectrum estimate given by $\{\hat{\beta}(\omega_k)\}_{k=1}^{K}$. If desired, this estimate can be transformed into a power spectrum estimate, as explained in Complement 5.6.4.

A cyclic minimizer reduces the criterion function at each iteration, as discussed in Complement 4.9.5. Furthermore, in the present case, this reduction is strict, because the solutions to the minimization problems with respect to y_u and to $\{h(\omega_k), \beta(\omega_k)\}$ in Steps 1 and 2 are unique under weak conditions. Combining this observation with the fact that the criterion in (5.6.56) is bounded from below by zero, we can conclude that the GAPES algorithm converges to a minimum point of (5.6.56). This minimum may be a local or global minimum, depending in part on the quality of the initial estimates of $\{h(\omega_k), \beta(\omega_k)\}$ used in Step 0. This initialization step and the remaining steps in the GAPES algorithm are discussed in more detail next.

Step 0. A simple way to obtain initial estimates of $\{h(\omega_k), \beta(\omega_k)\}$ is to apply APES to the full-data sequence with $y_u = 0$. This way of initializing GAPES can be interpreted as permuting Step 1 with Step 2 in the algorithm and initializing the algorithm in Step 0 with $y_u = 0$.

A more elaborate initialization scheme consists of using only the available data samples to build the sample-covariance matrix $\hat{R}$ in (5.6.46) needed in APES. Provided that there are enough samples so that the resulting $\hat{R}$ matrix is nonsingular, this initialization scheme usually gives more accurate estimates of $\{h(\omega_k), \beta(\omega_k)\}$ than the ones obtained by setting $y_u = 0$. (See [Stoica, Larsson, and Li 2000] for details.)

Step 1. We want to find the solution $\hat{y}_u$ to the problem

$$\min_{y_u} \sum_{k=1}^{K} \sum_{t=m+1}^{N} \left| \hat{h}^*(\omega_k)\tilde{y}(t) - \hat{\beta}(\omega_k)e^{i\omega_k t} \right|^2 \tag{5.6.57}$$

where $\tilde{y}(t) = \left[y(t), y(t-1), \ldots, y(t-m)\right]^T$. We will show that this minimization problem is quadratic in y_u (for given $\{\hat{h}(\omega_k)\}$ and $\{\hat{\beta}(\omega_k)\}$) and thus admits of a closed-form solution.

Let $\hat{h}^*(\omega_k) = \left[h_{0,k}, h_{1,k}, \ldots, h_{m,k}\right]$, and define

$$H_k = \begin{bmatrix} h_{0,k} & h_{1,k} & \cdots & h_{m,k} & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & h_{0,k} & h_{1,k} & \cdots & h_{m,k} \end{bmatrix}, \qquad (N-m) \times N$$

$$\mu_k = \hat{\beta}(\omega_k) \begin{bmatrix} e^{i\omega_k N} \\ \vdots \\ e^{i\omega_k (m+1)} \end{bmatrix}, \qquad (N-m) \times 1$$

Using this notation, we can write the quadratic criterion in (5.6.57) as

$$\sum_{k=1}^{K} \left\| H_k \begin{bmatrix} y(N) \\ \vdots \\ y(1) \end{bmatrix} - \mu_k \right\|^2 \tag{5.6.58}$$

Next, we define the matrices A_k and U_k via the following equality:

$$H_k \begin{bmatrix} y(N) \\ \vdots \\ y(1) \end{bmatrix} = A_k y_a + U_k y_u \tag{5.6.59}$$

With this notation, the criterion in (5.6.58) becomes

$$\sum_{k=1}^{K} \| U_k y_u - (\mu_k - A_k y_a) \|^2 \tag{5.6.60}$$

The minimizer of (5.6.60) with respect to y_u is readily found (see Result R32 in Appendix A) to be

$$\hat{y}_u = \left[\sum_{k=1}^{K} U_k^* U_k \right]^{-1} \left[\sum_{k=1}^{K} U_k^* (\mu_k - A_k y_a) \right] \tag{5.6.61}$$

The preceding inverse matrix exists under weak conditions; for details, see [Stoica, Larsson, and Li 2000].

Step 2. The solution to this step can be computed by applying the APES algorithm in Complement 5.6.4 to the data sequence made from y_a and $\hat{y}_u$.

The description of the GAPES algorithm is now complete. Numerical experience with this algorithm, reported in [Stoica, Larsson, and Li 2000], suggests that GAPES has good performance, particularly for data consisting of a mixture of sinusoidal signals superimposed in noise.

5.6.6 Extensions of Filter-Bank Approaches to Two-Dimensional Signals

The following filter-bank approaches for one-dimensional (1D) signals have been discussed so far in this chapter and its complements:

- the periodogram,
- the refined filter bank method,

- the Capon method, and
- the APES method

In this complement, we will explain briefly how these *nonparametric* spectral analysis methods can be extended to the case of two-dimensional (2D) signals. In the process, we also provide new interpretations for some of these methods, which are particularly useful when we want very simple (although somewhat heuristic) derivations of the methods in question. We will in turn discuss the extension of each of the methods listed above. Note that 2D spectral analysis finds applications in image processing, synthetic aperture radar imagery, etc. See [Larsson, Li, and Stoica 2003] for a review that covers the 2D methods discussed in this complement and their application to synthetic aperture radar. The 2D extensions of some *parametric* methods for line-spectral analysis are discussed in Complement 4.9.7.

Periodogram

The 1D periodogram can be obtained by a least-squares (LS) fitting of the data $\{y(t)\}$ to a generic 1D sinusoidal sequence $\{\beta e^{i\omega t}\}$:

$$\min_{\beta} \sum_{t=1}^{N} \left| y(t) - \beta e^{i\omega t} \right|^2 \tag{5.6.62}$$

The solution to (5.6.62) is readily found to be

$$\beta(\omega) = \frac{1}{N} \sum_{t=1}^{N} y(t) e^{-i\omega t} \tag{5.6.63}$$

The squared modulus of (5.6.63) (scaled by N; see Section 5.2) gives the 1D periodogram

$$\frac{1}{N} \left| \sum_{t=1}^{N} y(t) e^{-i\omega t} \right|^2 \tag{5.6.64}$$

In the 2D case, let $\{y(t, \bar{t})\}$ (for $t = 1, \ldots, N$ and $\bar{t} = 1, \ldots, \bar{N}$) denote the available data matrix, and let $\{\beta e^{i(\omega t + \bar{\omega}\bar{t})}\}$ denote a generic 2D sinusoid. The LS fit of the data to the generic sinusoid

$$\min_{\beta} \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} \left| y(t, \bar{t}) - \beta e^{i(\omega t + \bar{\omega}\bar{t})} \right|^2 \iff \min_{\beta} \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} \left| y(t, \bar{t}) e^{-i(\omega t + \bar{\omega}\bar{t})} - \beta \right|^2 \tag{5.6.65}$$

has the following solution:

$$\beta(\omega, \bar{\omega}) = \frac{1}{N\bar{N}} \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} y(t, \bar{t}) e^{-i(\omega t + \bar{\omega}\bar{t})} \tag{5.6.66}$$

Much as in the 1D case, the scaled squared magnitude of (5.6.66) yields the *2D periodogram*

$$\boxed{\frac{1}{N\bar{N}}\left|\sum_{t=1}^{N}\sum_{\bar{t}=1}^{\bar{N}} y(t,\bar{t})e^{-i(\omega t+\bar{\omega}\bar{t})}\right|^2} \tag{5.6.67}$$

which can be computed efficiently by means of a 2D FFT algorithm, as described next.

The 2D FFT algorithm computes the 2D DTFT of a sequence $\{y(t,\bar{t})\}$ (for $t = 1,\ldots,N$; $\bar{t} = 1,\ldots,\bar{N}$) on a grid of frequency values defined by

$$\begin{aligned}\omega_k &= \frac{2\pi k}{N}, \qquad k = 0,\ldots,N-1\\ \bar{\omega}_\ell &= \frac{2\pi \ell}{\bar{N}}, \qquad \ell = 0,\ldots,\bar{N}-1\end{aligned}$$

The 2D FFT algorithm achieves computational efficiency by making use of the 1D FFT described in Section 2.3. Let

$$\begin{aligned}Y(k,\ell) &= \sum_{t=1}^{N}\sum_{\bar{t}=1}^{\bar{N}} y(t,\bar{t})e^{-i\left(\frac{2\pi k}{N}t+\frac{2\pi\ell}{\bar{N}}\bar{t}\right)}\\ &= \sum_{t=1}^{N} e^{-i\frac{2\pi k}{N}t}\underbrace{\sum_{\bar{t}=1}^{\bar{N}} y(t,\bar{t})e^{-i\frac{2\pi\ell}{\bar{N}}\bar{t}}}_{\triangleq V_t(\ell)} &\quad(5.6.68)\\ &= \sum_{t=1}^{N} V_t(\ell)e^{-i\frac{2\pi k}{N}t} &\quad(5.6.69)\end{aligned}$$

For each $t = 1,\ldots,N$, the sequence $\{V_t(\ell)\}_{\ell=0}^{\bar{N}-1}$ defined in (5.6.68) can be efficiently computed by using a 1D FFT of length $\bar{N}$ (*cf.* Section 2.3). In addition, for each $\ell = 0,\ldots,\bar{N}-1$, the sum in (5.6.69) can be efficiently computed by using a 1D FFT of length N. If N is a power of two, an N-point 1D FFT requires $\frac{N}{2}\log_2 N$ flops. Thus, if N and $\bar{N}$ are powers of two, then the number of operations needed to compute $\{Y(k,\ell)\}$ is

$$\boxed{N\frac{\bar{N}}{2}\log_2\bar{N} + \bar{N}\frac{N}{2}\log_2 N = \frac{N\bar{N}}{2}\log_2(N\bar{N}) \text{ flops}} \tag{5.6.70}$$

If N or $\bar{N}$ is not a power of two, zero padding can be used.

Refined Filter-Bank (RFB) Method

Similarly to the 1D case (see (5.3.30) or (5.7.1)), the *2D RFB method* can be implemented as a multiwindowed periodogram (*cf.* (5.6.67)):

$$\boxed{\frac{1}{K}\sum_{p=1}^{K}\left|\sum_{t=1}^{N}\sum_{\bar{t}=1}^{\bar{N}} w_p(t,\bar{t})\, y(t,\bar{t})\, e^{-i(\omega t+\bar{\omega}\bar{t})}\right|^2} \tag{5.6.71}$$

where $\{w_p(t,\bar{t})\}_{p=1}^{K}$ are the 2D Slepian data windows (or tapers). The problem left is to derive 2D extensions of the 1D Slepian tapers, discussed in Section 5.3.1.

The frequency response of a 2D taper $\{w(t,\bar{t})\}$ is given by

$$\sum_{t=1}^{N}\sum_{\bar{t}=1}^{\bar{N}} w(t,\bar{t})e^{-i(\omega t+\bar{\omega}\bar{t})} \tag{5.6.72}$$

Let us define the matrices

$$W = \begin{bmatrix} w(1,1) & \cdots & w(1,\bar{N}) \\ \vdots & & \vdots \\ w(N,1) & \cdots & w(N,\bar{N}) \end{bmatrix}$$

$$B = \begin{bmatrix} e^{-i(\omega+\bar{\omega})} & \cdots & e^{-i(\omega+\bar{\omega}\bar{N})} \\ \vdots & & \vdots \\ e^{-i(\omega N+\bar{\omega})} & \cdots & e^{-i(\omega N+\bar{\omega}\bar{N})} \end{bmatrix}$$

and let vec$(\cdot)$ denote the vectorizaton operator that stacks the columns of its matrix argument into a single vector. Also, let

$$a(\omega) = \begin{bmatrix} e^{-i\omega} \\ \vdots \\ e^{-iN\omega} \end{bmatrix}, \qquad \bar{a}(\omega) = \begin{bmatrix} e^{-i\bar{\omega}} \\ \vdots \\ e^{-i\bar{N}\bar{\omega}} \end{bmatrix} \tag{5.6.73}$$

and let the symbol $\otimes$ denote the Kronecker matrix product. (The Kronecker product of two matrices, X of size $m \times n$ and Y of size $\bar{m} \times \bar{n}$, is an $m\bar{m} \times n\bar{n}$ matrix whose (i,j) block of size $\bar{m} \times \bar{n}$ is given by $X_{ij} \cdot Y$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, where X_{ij} denotes the (i,j)th element of X; see, e.g., [Horn and Johnson 1985] for the properties of $\otimes$.) Finally, let

$$\begin{aligned} w &= \text{vec}(W) \\ &= \left[w(1,1),\ldots,w(N,1)|\cdots|w(1,\bar{N}),\ldots,w(N,\bar{N})\right]^T \end{aligned} \tag{5.6.74}$$

and

$$\begin{aligned} b(\omega, \bar{\omega}) &= \operatorname{vec}(B) \\ &= \left[e^{-i(\omega+\bar{\omega})}, \ldots, e^{-i(\omega N+\bar{\omega})} | \cdots | e^{-i(\omega+\bar{\omega}\bar{N})}, \ldots, e^{-i(\omega N+\bar{\omega}\bar{N})}\right]^T \\ &= \bar{a}(\bar{\omega}) \otimes a(\omega) \end{aligned} \tag{5.6.75}$$

(The last equality in (5.6.75) follows from the definition of ⊗.) Using (5.6.74) and (5.6.75), we can write (5.6.72) as

$$w^* b(\omega, \bar{\omega}) \tag{5.6.76}$$

which is similar to the expression $h^* a(\omega)$ for the 1D frequency response in Section 5.3.1. Hence, the analysis in Section 5.3.1 carries over to the 2D case, with the only difference being that now the matrix Γ is given by

$$\begin{aligned} \Gamma_{2\text{D}} &= \frac{1}{(2\pi)^2} \int_{-\beta\pi}^{\beta\pi} \int_{-\bar{\beta}\pi}^{\bar{\beta}\pi} b(\omega, \bar{\omega}) b^*(\omega, \bar{\omega}) d\omega\, d\bar{\omega} \\ &= \frac{1}{(2\pi)^2} \int_{-\beta\pi}^{\beta\pi} \int_{-\bar{\beta}\pi}^{\bar{\beta}\pi} \left[\bar{a}(\bar{\omega})\bar{a}^*(\bar{\omega})\right] \otimes \left[a(\omega)a^*(\omega)\right] d\omega\, d\bar{\omega} \end{aligned}$$

where we have used the fact that $(A \otimes B)(C \otimes D) = AC \otimes BD$ for any conformable matrices. (See, for example, [Horn and Johnson 1985].) Hence,

$$\boxed{\Gamma_{2\text{D}} = \bar{\Gamma}_{1\text{D}} \otimes \Gamma_{1\text{D}}} \tag{5.6.77}$$

where

$$\Gamma_{1\text{D}} = \frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} a(\omega)a^*(\omega) d\omega, \qquad \bar{\Gamma}_{1\text{D}} = \frac{1}{2\pi} \int_{-\bar{\beta}\pi}^{\bar{\beta}\pi} \bar{a}(\bar{\omega})\bar{a}^*(\bar{\omega}) d\bar{\omega} \tag{5.6.78}$$

The preceding Kronecker-product expression of $\Gamma_{2\text{D}}$ implies (see [Horn and Johnson 1985]) that

(a) the eigenvalues of $\Gamma_{2\text{D}}$ are equal to the products of the eigenvalues of $\Gamma_{1\text{D}}$ and $\bar{\Gamma}_{1\text{D}}$; and
(b) the eigenvectors of $\Gamma_{2\text{D}}$ are given by the Kronecker products of the eigenvectors of $\Gamma_{1\text{D}}$ and $\bar{\Gamma}_{1\text{D}}$.

The conclusion is that *the computation of 2D Slepian tapers can be reduced to the computation of 1D Slepian tapers*. We refer the reader to Section 5.3.1 for details on computation of 1D Slepian tapers.

Capon and APES Methods

In the 1D case, we can obtain the Capon and APES methods by a weighted LS fit of the data vectors $\{\tilde{y}(t)\}$, where

$$\tilde{y}(t) = \left[y(t), y(t-1), \ldots, y(t-m)\right]^T \tag{5.6.79}$$

to the vectors corresponding to a generic sinusoidal signal with frequency ω. Specifically, consider this LS problem:

$$\min_{\beta} \sum_{t=m+1}^{N} \left[\tilde{y}(t) - a(\omega)\beta e^{i\omega t}\right]^* W^{-1} \left[\tilde{y}(t) - a(\omega)\beta e^{i\omega t}\right] \tag{5.6.80}$$

where W^{-1} is a weighting matrix that is yet to be specified and where

$$a(\omega) = \left[1, e^{-i\omega}, \ldots, e^{-im\omega}\right]^T \tag{5.6.81}$$

Note that the definition of $a(\omega)$ in (5.6.81) differs from that of $a(\omega)$ in (5.6.73). The solution to (5.6.80) is given by

$$\beta(\omega) = \frac{a^*(\omega)W^{-1}g(\omega)}{a^*(\omega)W^{-1}a(\omega)} \tag{5.6.82}$$

where

$$g(\omega) = \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t)e^{-i\omega t} \tag{5.6.83}$$

For

$$W = \hat{R} \triangleq \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t)\tilde{y}^*(t) \tag{5.6.84}$$

the weighted LS estimate of the amplitude spectrum in (5.6.82) reduces to the Capon method (see equation (5.6.54) in Complement 5.6.4), whereas for

$$W = \hat{R} - g(\omega)g^*(\omega) \triangleq \hat{Q}(\omega) \tag{5.6.85}$$

equation (5.6.82) gives the APES method. (See equations (5.6.47), (5.6.48), and (5.6.50) in Complement 5.6.4.)

The extension of the above derivation to the 2D case is straightforward. By analogy with the 1D data vector in (5.6.79), let

$$\left[y(t-k, \bar{t}-\bar{k})\right] = \begin{bmatrix} y(t,\bar{t}) & \cdots & y(t,\bar{t}-\bar{m}) \\ \vdots & & \vdots \\ y(t-m,\bar{t}) & \cdots & y(t-m,\bar{t}-\bar{m}) \end{bmatrix} \tag{5.6.86}$$

be the 2D data matrix, and let

$$\begin{aligned}\tilde{y}(t,\bar{t}) &= \operatorname{vec}\left(\left[y(t-k,\bar{t}-\bar{k})\right]\right)\\ &= \left[y(t,\bar{t}),\ldots,y(t-m,\bar{t})|\cdots|y(t,\bar{t}-\bar{m}),\ldots,y(t-m,\bar{t}-\bar{m})\right]^T\end{aligned} \tag{5.6.87}$$

Our goal is to fit the data matrix in (5.6.86) to the matrix corresponding to a generic 2D sinusoid with frequency pair $(\omega,\bar{\omega})$—that is,

$$\left[\beta e^{i[\omega(t-k)+\bar{\omega}(\bar{t}-\bar{k})]}\right] = \beta\begin{bmatrix} e^{i[\omega t+\bar{\omega}\bar{t}]} & \cdots & e^{i[\omega t+\bar{\omega}(\bar{t}-\bar{m})]}\\ \vdots & & \vdots\\ e^{i[\omega(t-m)+\bar{\omega}\bar{t}]} & \cdots & e^{i[\omega(t-m)+\bar{\omega}(\bar{t}-\bar{m})]}\end{bmatrix} \tag{5.6.88}$$

As with (5.6.87), let us vectorize (5.6.88):

$$\begin{aligned}\operatorname{vec}\left(\left[\beta e^{i[\omega(t-k)+\bar{\omega}(\bar{t}-\bar{k})]}\right]\right) &= \beta e^{i(\omega t+\bar{\omega}\bar{t})}\operatorname{vec}\left(\left[e^{-i(\omega k+\bar{\omega}\bar{k})}\right]\right)\\ &= \beta e^{i(\omega t+\bar{\omega}\bar{t})}\bar{a}(\bar{\omega})\otimes a(\omega)\end{aligned} \tag{5.6.89}$$

As in (5.6.75), let

$$\boxed{b(\omega,\bar{\omega}) = \bar{a}(\bar{\omega})\otimes a(\omega), \qquad (m+1)(\bar{m}+1)\times 1} \tag{5.6.90}$$

We deduce from (5.6.87)–(5.6.90) that the 2D counterpart of the 1D weighted-LS fitting problem in (5.6.80) is the following:

$$\begin{aligned}\min_{\beta}\sum_{t=m+1}^{N}\sum_{\bar{t}=\bar{m}+1}^{\bar{N}} &\left[\tilde{y}(t,\bar{t})-\beta e^{i(\omega t+\bar{\omega}\bar{t})}b(\omega,\bar{\omega})\right]^* W^{-1}\\ &\cdot\left[\tilde{y}(t,\bar{t})-\beta e^{i(\omega t+\bar{\omega}\bar{t})}b(\omega,\bar{\omega})\right]\end{aligned} \tag{5.6.91}$$

The solution to (5.6.91) is given by

$$\boxed{\beta(\omega,\bar{\omega}) = \frac{b^*(\omega,\bar{\omega})W^{-1}g(\omega,\bar{\omega})}{b^*(\omega,\bar{\omega})W^{-1}b(\omega,\bar{\omega})}} \tag{5.6.92}$$

where

$$\boxed{g(\omega,\bar{\omega}) = \frac{1}{(N-m)(\bar{N}-\bar{m})}\sum_{t=m+1}^{N}\sum_{\bar{t}=\bar{m}+1}^{\bar{N}}\tilde{y}(t,\bar{t})e^{-i(\omega t+\bar{\omega}\bar{t})}} \tag{5.6.93}$$

The *2D Capon method* is given by (5.6.92) with

$$\boxed{W = \frac{1}{(N-m)(\bar{N}-\bar{m})} \sum_{t=m+1}^{N} \sum_{\bar{t}=\bar{m}+1}^{\bar{N}} \tilde{y}(t,\bar{t})\tilde{y}^{*}(t,\bar{t}) \triangleq \hat{R}} \tag{5.6.94}$$

whereas the *2D APES method* is given by (5.6.92) with

$$\boxed{W = \hat{R} - g(\omega,\bar{\omega})g^{*}(\omega,\bar{\omega}) \triangleq \hat{Q}(\omega,\bar{\omega})} \tag{5.6.95}$$

Note that $g(\omega,\bar{\omega})$ in (5.6.93) can be evaluated efficiently by use of a 2D FFT algorithm. However, an efficient implementation of the 2D spectral estimate in (5.6.92) is not so direct. A naive implementation could be rather time consuming, as a result of the large dimensions of the vectors and matrices involved and of the need to evaluate $\beta(\omega,\bar{\omega})$ on a 2D frequency grid. We refer the reader to [Larsson, Li, and Stoica 2003] and the references therein for a discussion of computationally efficient implementations of 2D Capon and 2D APES spectral estimation methods.

5.7 EXERCISES

Exercise 5.1: Multiwindow Interpretation of Bartlett and Welch Methods

Equation (5.3.30) allows us to interpret the RFB method as a *multiwindow* (or *multitaper*) approach. Indeed, according to equation (5.3.30), we can write the RFB spectral estimator as

$$\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} w_{p,t} y(t) e^{-i\omega t} \right|^{2} \tag{5.7.1}$$

where K is the number of data windows (or tapers) and where, in the case of RFB, the $w_{p,t}$ are obtained from the pth dominant Slepian sequence ($p = 1, \ldots, K$).

Show that the Bartlett and Welch methods can also be cast into the preceding multiwindow framework. Make use of the multiwindow interpretation of these methods to compare them with one another and with the RFB approach.

Exercise 5.2: An Alternative Statistically Stable RFB Estimate

In Section 5.3.3, we developed a statistically stable RFB spectral estimator, using a bank of narrow bandpass filters. In Section 5.4, we derived the Capon method, which employs a shorter filter length than does the RFB. In this exercise, we derive the RFB analog of the Capon approach and show its correspondence with the Welch and the Blackman–Tukey estimators.

Instead of the filter in (5.3.4), consider a passband filter of shorter length:

$$h = [h_0, \ldots, h_m]^{*} \tag{5.7.2}$$

for some $m < N$. The optimal h will be the first Slepian sequence in (5.3.10) found by using a Γ matrix of size $m \times m$. In this case, the filtered output

$$y_F(t) = \sum_{k=0}^{m} h_k \tilde{y}(t-k) \tag{5.7.3}$$

(with $\tilde{y}(t) = y(t)e^{-i\omega t}$) can be computed for $t = m+1, \ldots, N$. The resulting RFB spectral estimate is given by

$$\hat{\phi}(\omega) = \frac{1}{N-m} \sum_{t=m+1}^{N} |y_F(t)|^2 \tag{5.7.4}$$

- **(a)** Show that the estimator in (5.7.4) is an unbiased estimate of $\phi(\omega)$, under the standard assumptions considered in this chapter.
- **(b)** Show that $\hat{\phi}(\omega)$ can be written as

$$\hat{\phi}(\omega) = \frac{1}{m+1} h^*(\omega)\hat{R}\, h(\omega) \tag{5.7.5}$$

  where $\hat{R}$ is an $(m+1) \times (m+1)$ Hermitian (but not necessarily Toeplitz) estimate of the covariance matrix of $y(t)$. Find the corresponding filter $h(\omega)$.
- **(c)** Compare (5.7.5) with the Blackman–Tukey estimate in equation (5.4.22). Discuss how the two compare when N is large.
- **(d)** Interpret $\hat{\phi}(\omega)$ as a Welch-type estimator. What is the overlap parameter K in the corresponding Welch method?

Exercise 5.3: Another Derivation of the Capon FIR Filter

The Capon FIR-filter design problem can be restated as follows:

$$\min_h \; h^*Rh/|h^*a(\omega)|^2 \tag{5.7.6}$$

Make use of the Cauchy–Schwartz inequality (Result R22 in Appendix A) to obtain a simple proof of the fact that the h given by (5.4.8) is a solution to this optimization problem.

Exercise 5.4: The Capon Filter Is a Matched Filter

Compare the Capon filter-design problem (5.4.7) with the following classical *matched filter design*.

- Filter: A causal FIR filter with an $(m+1)$-dimensional impulse-response vector, denoted by h.
- Signal-in-noise model: $y(t) = \alpha e^{i\omega t} + \varepsilon(t)$, which gives the following expression for the input vector to the filter:

$$z(t) = \alpha a(\omega)e^{i\omega t} + e(t) \tag{5.7.7}$$

where $a(\omega)$ is as defined in (5.4.6), $\alpha e^{i\omega t}$ is a sinusoidal signal,

$$z(t) = [y(t), y(t-1), \ldots, y(t-m)]^T$$

and $e(t)$ is a possibly colored noise vector defined similarly to $z(t)$. The preceding signal and noise terms are assumed to be uncorrelated.

- Design goal: Maximize the signal-to-noise ratio at the filter's output,

$$\max_h |h^* a(\omega)|^2 / h^* Q h \tag{5.7.8}$$

where Q is the noise-covariance matrix.

Show that the Capon filter is identical to the matched filter that solves this design problem. The adjective "matched" attached to the preceding filter is motivated by the fact that the filter impulse response vector h depends on, and hence is "matched to," the signal term in (5.7.7).

Exercise 5.5: Computation of the Capon Spectrum

The Capon spectral estimators are defined in equations (5.4.19) and (5.4.20). The bulk of the computation of either estimator consists in the evaluation of an expression of the form $a^*(\omega)Qa(\omega)$, where Q is a given positive definite matrix, at a number of points on the frequency axis. Let these evaluation points be given by $\{\omega_k = 2\pi k/M\}_{k=0}^{M-1}$ for some sufficiently large M value (which we assume to be a power of two). The direct evaluation of $a^*(\omega_k)Qa(\omega_k)$, for $k = 0, \ldots, M-1$, would require $\mathcal{O}(Mm^2)$ flops. Show that an evaluation based on the eigendecomposition of Q and the use of FFT is usually much more efficient computationally.

Exercise 5.6: A Relationship between the Capon Method and MUSIC (Pseudo)Spectra

Assume that the covariance matrix R, entering the Capon spectrum formula, has the expression (4.2.7) in the frequency-estimation application. Then, show that

$$\lim_{\sigma^2 \to 0} (\sigma^2 R^{-1}) = I - A(A^*A)^{-1}A^* \tag{5.7.9}$$

Conclude that the limiting (for $N \gg 1$) Capon and MUSIC (pseudo)spectra, associated with the frequency-estimation data, are close to one another, provided that all signal-to-noise ratios are large enough.

Exercise 5.7: A Capon-Like Implementation of MUSIC

The Capon and MUSIC (pseudo)spectra, as the data length N increases, are given by the functions in equations (5.4.12) and (4.5.13), respectively. Recall that the columns of the matrix G in (4.5.13) are equal to the $(m-n)$ eigenvectors corresponding to the smallest eigenvalues of the covariance matrix R in (5.4.12).

Consider the Capon-like pseudospectrum

$$g_k(\omega) = a^*(\omega)R^{-k}a(\omega)\lambda^k \tag{5.7.10}$$

where λ is the minimum eigenvalue of R; the covariance matrix R is assumed to have the form (4.2.7) postulated by MUSIC. Show that, under this assumption,

$$\boxed{\lim_{k\to\infty} g_k(\omega) = a^*(\omega)GG^*a(\omega) = (4.5.13)} \tag{5.7.11}$$

(where the convergence is uniform in ω). Explain why the convergence in (5.7.11) could be slow in difficult scenarios, such as those with closely spaced frequencies, and hence that the use of (5.7.10) with a large k to approximate the MUSIC pseudospectrum could be computationally inefficient. However, the use of (5.7.10) for frequency estimation has a potential advantage over MUSIC that might outweigh its computational inefficiency. Find and comment on that advantage.

Exercise 5.8: Capon Estimate of the Parameters of a Single Sine Wave

Assume that the data under study consists of a sinusoidal signal observed in white noise. In such a case, the covariance matrix R is given (*cf.* (4.2.7)) by

$$R = \alpha^2 a(\omega_0)a(\omega_0)^* + \sigma^2 I, \qquad (m \times m)$$

where ω_0 denotes the true frequency value. Show that the limiting (as $N \to \infty$) Capon spectrum (5.4.12) peaks at $\omega = \omega_0$. Derive the height of the peak and show that it is not equal to α^2 (as might have been expected), but is given by a function of α^2, m, and σ^2. Conclude that the Capon method can be used to obtain a consistent estimate of the frequency of a *single* sinusoidal signal in white noise (but not of the signal power).

We note that, for two or more sinusoidal signals, the Capon frequency estimates are inconsistent. Hence, the Capon frequency estimator behaves somewhat like the AR frequency estimation method in this respect; see Exercise 4.4.

Exercise 5.9: An Alternative Derivation of the Relationship between the Capon and AR Methods

Make use of equation (3.9.17), relating R_{m+1}^{-1}, to R_m^{-1}, to obtain a simple proof of the formula (5.4.36) relating the Capon and AR spectral estimators.

COMPUTER EXERCISES

Tools for Filter-Bank Spectral Estimation:

The text website `www.prenhall.com/stoica` contains the following MATLAB functions for use in computing filter-bank spectral estimates.

- `h=slepian(N,K,J)`
  Returns the first `J` Slepian sequences, given `N` and `K`, as defined in Section 5.3; `h` is an $N \times J$ matrix whose ith column gives the ith Slepian sequence.

- `phi=rfb(y,K,L)`
  The RFB spectral estimator. The vector `y` is the input data vector, `L` controls the frequency-sample spacing of the output, and the output vector `phi`$= \hat{\phi}(\omega_k)$, where $\omega_k = \frac{2\pi k}{L}$. For $K = 1$, this function implements the high-resolution RFB method in equation (5.3.22); for $K > 1$, it implements the statistically stable RFB method.
- `phi=capon(y,m,L)`
  The CM Version-1 spectral estimator in equation (5.4.19); `y`, `L`, and `phi` are as for the RFB spectral estimator, and `m` is the size of the square matrix $\hat{R}$.

Exercise C5.10: Slepian Window Sequences
We consider the Slepian window sequences for both $K = 1$ (high resolution) and $K = 4$ (lower resolution, higher statistical stability) and compare them with classical window sequences.

(a) Evaluate and plot the first 8 Slepian window sequences and their Fourier transforms for $K = 1$ and 4 and for $N = 32$, 64, and 128 (and perhaps other values, too). Qualitatively describe the filter passbands of these first 8 Slepian sequences for $K = 1$ and $K = 4$. Which act as lowpass filters and which act as “other” types of filters?

(b) In this chapter, we showed that, for “large N” and $K = 1$, the first Slepian sequence is “reasonably close to” the rectangular window; compare the first Slepian sequence and its Fourier transform for $N = 32$, 64, and 128 to the rectangular window and its Fourier transform. How do they compare as a function of N? In the light of this comparison, how do you expect the high-resolution RFB PSD estimator to perform relative to the periodogram?

Exercise C5.11: Resolution of Refined Filter-Bank Methods
We will compare the resolving power of the RFB spectral estimator with $K = 1$ to that of the periodogram. To do so, we look at the spectral estimates of sequences that are made up of two sinusoids in noise, where we vary the frequency difference.

Generate the sequences

$$y_\alpha(t) = 10\sin(0.2 \cdot 2\pi t) + 5\sin((0.2 + \alpha/N)2\pi t)$$

for various values of α near 1. Compare the resolving ability of the RFB power spectral estimate for $K = 1$ and of the periodogram for both $N = 32$ and $N = 128$. Discuss your results in relation to the theoretical comparisons between the two estimators. Do the results echo the theoretical predictions based on the analysis of Slepian sequences?

Exercise C5.12: The Statistically Stable RFB Power Spectral Estimator
In this exercise, we will compare the RFB power spectral estimator when $K = 4$ to the Blackman–Tukey and Daniell estimators. We will use the narrowband and broadband processes considered in Exercise C2.22.

Broadband ARMA Process.

(a) Generate 50 realizations of the broadband ARMA process in Exercise C2.22, using $N = 256$. Estimate the spectrum by using

- The RFB method with $K = 4$.
- The Blackman–Tukey method with an appropriate window (such as the Bartlett window) and window length M. Choose M to obtain performance similar that of the RFB method. (You can select an appropriate value of M off-line and verify it in your experiments.)
- The Daniell method with $\tilde{N} = 8N$ and an appropriate choice of J. Choose J to obtain performance similar to that of the RFB method. (You can select J off-line and verify it in your experiments.)

(b) Evaluate the relative performance of the three estimators in terms of bias and variance. Are the comparisons in agreement with the theoretical predictions?

Narrowband ARMA Process. Repeat parts (a) and (b) using 50 realizations (with $N = 256$) of the narrowband ARMA process in Exercise C2.22.

Exercise C5.13: The Capon Method

In this exercise, we compare the Capon method to the RFB and AR methods. Consider the sinusoidal data sequence in equation (2.9.20) from Exercise C2.19, with $N = 64$.

(a) We first compare the data filters corresponding to a RFB method (in which the filter is data independent) with the filter corresponding to the CM Version-1 method, using both $m = N/4$ and $m = N/2 - 1$; we choose the Slepian RFB method with $K = 1$ and $K = 4$ for this comparison. For two estimation frequencies, $\omega = 0$ and $\omega = 2\pi \cdot 0.1$, plot the frequency response of the five filters (1 for $K = 1$ and 4 for $K = 4$), shown in the first block of Figure 5.1 for the two RFB methods, and also plot the response of the two Capon filters (one for each value of m; see (5.4.5) and (5.4.8)). What are their characteristic features in relation to the data? Using these plots, discuss how data dependence can improve spectral estimation performance.

(b) Compare the two Capon estimators with the RFB estimator for both $K = 1$ and $K = 4$. Generate 50 Monte Carlo realizations of the data, and overlay plots of the 50 spectral estimates for each estimator. Discuss the similarities and differences between the RFB and Capon estimators.

(c) Compare Capon and least-squares AR spectral estimates, again by generating 50 Monte Carlo realizations of the data and overlaying plots of the 50 spectral estimates. Use $m = 8$, 16, and 30 for both the Capon method and the AR model order. How do the two methods compare in terms of resolution and variance? What are your main summarizing conclusions? Explain your results in terms of the data characteristics.

6

Spatial Methods

6.1 INTRODUCTION

In this chapter, we consider the problem of *locating n radiating sources by using an array of m passive sensors*, as shown in Figure 6.1. The emitted energy from the sources could be, for example, acoustic or electromagnetic, and the receiving sensors could be any transducers that convert the received energy to electrical signals. Examples of sensors include electromagnetic antennas, hydrophones, and seismometers. This type of problem finds applications in *radar and sonar systems, communications, astrophysics, biomedical research, seismology, underwater surveillance* (also called passive listening), and many other fields. This problem basically consists of determining how the "energy" is distributed over *space* (which can be air, water, or the earth), with the source positions representing points in space with high concentrations of energy. Hence, it can be named a *spatial spectral estimation problem.* This name is also motivated by the fact that there are close ties between the source-location problem and the problem of temporal spectral estimation, treated in Chapters 1–5. In fact, as we will see, almost any of the methods encountered in the previous chapters may be used to derive a solution for the source-location problem.

The emphasis in this chapter will be on *developing a model for the output signal of the receiving sensor array*. Once this model has been derived, the source-location problem is turned into a parameter-estimation problem that is quite similar to the temporal frequency-finding application discussed in Chapter 4. Hence, as we shall see, most of the methods developed for frequency estimation can be used to solve the spatial problem of source location.

The sources in Figure 6.1 generate a *wave field* that travels through space and is *sampled*, *in both space and time*, *by the sensor array*. By making an analogy with temporal sampling, we may expect that the spatial sampling done by the array provides more and more information on

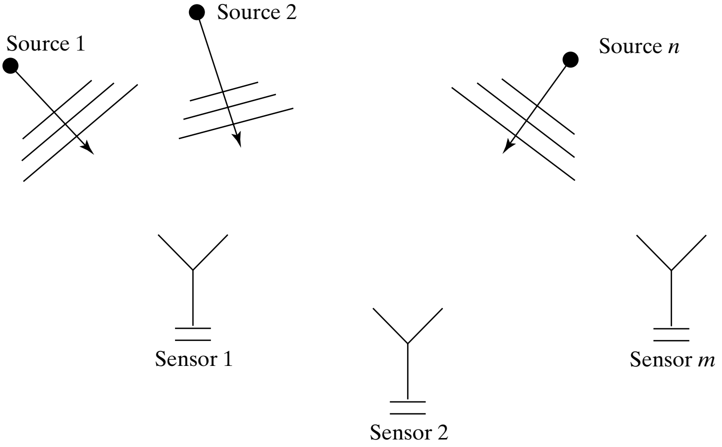


Figure 6.1 The set-up of the source-location problem.

the incoming waves as the *array's aperture* increases. The array's aperture is the space occupied by the array, as measured in units of signal wavelength. It is then no surprise that an array of sensors can provide significantly enhanced location performance as compared to the use of a *single antenna* (which was the system used in the early applications of the source-location problem.)

The development of the array model in the next section is based on a number of simplifying assumptions. Some of these assumptions, which have a more general character, are listed below. The sources are assumed to be situated in the *far field* of the array. Furthermore, we assume that both the sources and the sensors in the array are in the *same plane* and that the sources are *point emitters*. In addition, it is assumed that the *propagation medium is homogeneous (*i.e., *not dispersive)*, and so the waves arriving at the array can be considered to be *planar*. Under these assumptions, the only parameter that characterizes the source locations is the so-called *angle of arrival*, or *direction of arrival* (DOA); the DOA will be formally defined later on.

The above assumptions may be relaxed, but only at the expense of significantly complicating the array model. Note that, in the general case of a near-field source and a three-dimensional array, three parameters are required to define the position of one source—for instance the *azimuth*, *elevation*, and *range*. Nevertheless, if the assumption of planar waves is maintained, then we can treat the case of several unknown parameters per source without complicating the model too much. However, in order to keep the discussion as simple as possible, we will consider only the case of one parameter per source.

In this chapter, it is also assumed that *the number of sources n is known*. The selection of n, when it is unknown, is a problem of significant importance for many applications, which is often referred to as the *detection problem*. For solutions to the detection problem (which is analogous to the problem of order selection in signal modeling), the reader is referred to [Wax and Kailath 1985; Fuchs 1988; Viberg, Ottersten, and Kailath 1991; Fuchs 1992] and Appendix C.

Finally, it is assumed that the sensors in the array can be modeled as linear (time-invariant) systems, and that both their transfer characteristics and their locations are known. In short, we say that *the array is assumed to be calibrated*.

6.2 ARRAY MODEL

We begin by considering the case of a *single source*. Once we establish a model of the array for this case, the general model for the multiple-source case is simply obtained by the superposition principle.

Suppose that a single waveform impinges upon the array, and let $x(t)$ denote the value of the signal waveform as measured at some *reference point*, at time t. The "reference point" may be one of the sensors in the array or any other point placed near enough to the array so that the previously made assumption of planar wave propagation holds true. The physical signals received by the array are *continuous time waveforms*; hence, t is a continuous variable here, unless otherwise stated.

Let τ_k denote the time needed for the wave to travel from the reference point to sensor k ($k = 1, \dots, m$). Then the output of sensor k can be written as

$$\bar{y}_k(t) = \bar{h}_k(t) * x(t - \tau_k) + \bar{e}_k(t) \tag{6.2.1}$$

where $\bar{h}_k(t)$ is the impulse response of the kth sensor, "$*$" denotes the convolution operation, and $\bar{e}_k(t)$ is an *additive noise*. The noise may enter in equation (6.2.1) either as "thermal noise" generated by the sensor's circuitry, as "random background radiation" impinging on the array, or in other ways. In (6.2.1), $\bar{h}_k(t)$ is assumed known, but both the "input" signal $x(t)$ and the delay τ_k are unknown. The parameters characterizing the source location enter into (6.2.1) through $\{\tau_k\}$. Hence, the source-location problem is basically one of *time-delay estimation for the unknown input case*.

The model equation (6.2.1) can be simplified significantly if *the signals are assumed to be narrowband*. In order to show how this can be done, a number of preliminaries are required.

Let $X(\omega)$ denote the Fourier transform of the (continuous-time) signal $x(t)$:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt \tag{6.2.2}$$

(which is assumed to exist and be finite for all $\omega \in (-\infty, \infty)$). The inverse transform, which expresses $x(t)$ as a linear functional of $X(\omega)$, is given by

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{i\omega t}d\omega \tag{6.2.3}$$

Similarly, we define the transfer function $\bar{H}_k(\omega)$ of the kth sensor as the Fourier transform of $\bar{h}_k(t)$. In addition, let $\bar{Y}_k(\omega)$ and $\bar{E}_k(\omega)$ denote the Fourier transforms of the signal $\bar{y}_k(t)$ and noise $\bar{e}_k(t)$ in (6.2.1). By using this notation and the properties of the Fourier transform, $\bar{Y}_k(\omega)$ can be written as

$$\bar{Y}_k(\omega) = \bar{H}_k(\omega)X(\omega)e^{-i\omega\tau_k} + \bar{E}_k(\omega) \tag{6.2.4}$$

For a general class of physical signals, such as the carrier-modulated signals encountered in communications, the energy spectral density of $x(t)$ has the form shown in Figure 6.2. There, ω_c denotes the *center (or carrier) frequency*, which is usually the center of the frequency band occupied by the signal (hence its name). A signal having an energy spectrum of the form depicted in Figure 6.2 is called a *bandpass signal* (by direct analogy with the notion of bandpass filters).

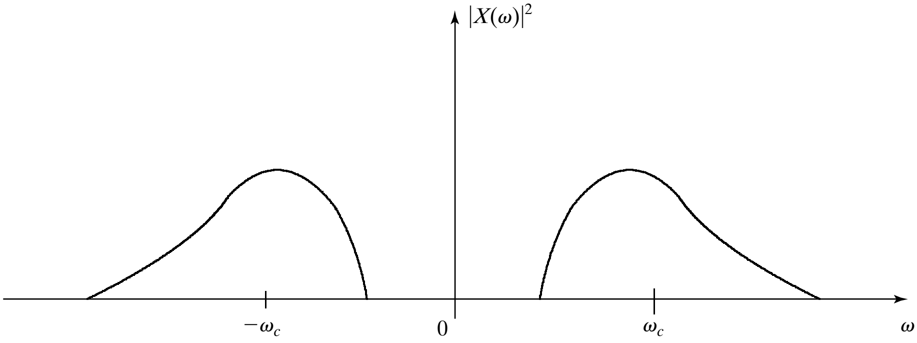


Figure 6.2 The energy spectrum of a bandpass signal.

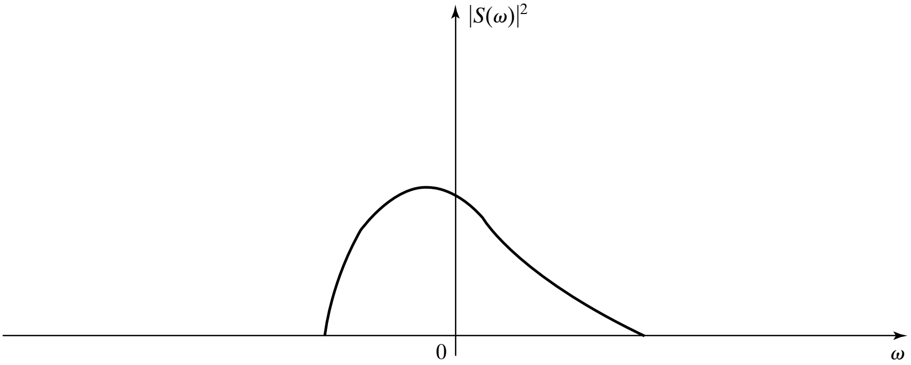


Figure 6.3 The baseband spectrum that gives rise to the bandpass spectrum in Figure 6.2.

For now, *we assume that the received signal $x(t)$ is bandpass*. It is clear from Figure 6.2 that the spectrum of such a signal is completely defined by the spectrum of a corresponding *baseband (or lowpass) signal*. The baseband spectrum—say, $|S(\omega)|^2$—corresponding to the one in Figure 6.2, is displayed in Figure 6.3. Let $s(t)$ denote the baseband signal associated with $x(t)$. The process of obtaining $x(t)$ from $s(t)$ is called *modulation*; the inverse process is called *demodulation*. In what follows, we make a number of comments on the modulation and demodulation processes, which—while not being strictly relevant to the source-location problem—could be helpful in clarifying some claims in the text.

6.2.1 The Modulation-Transmission-Demodulation Process

The physical signal $x(t)$ is real valued; hence, its spectrum $|X(\omega)|^2$ should be even (i.e., symmetric about $\omega = 0$; see, for instance, Figure 6.2). On the other hand, the spectrum of the demodulated signal $s(t)$ might not be even (as indicated in Figure 6.3); hence, $s(t)$ might be complex valued. The way in which this can happen is explained as follows. The *transmitted* signal is, of course, obtained by modulating a real-valued signal. Hence, in the spectrum of the transmitted signal, the baseband spectrum is symmetric about $\omega = \omega_c$. The characteristics of the *transmission channel* (or the *propagation medium*), however, most often are asymmetric about $\omega = \omega_c$. This results in a *received* bandpass signal with an associated baseband spectrum that is not even. Hence, the demodulated received signal is complex-valued. This observation supports a claim made in Chapter 1 that complex-valued signals are not uncommon in spectral estimation problems.

The Modulation Process. If $s(t)$ is multiplied by $e^{i\omega_c t}$, then the Fourier transform of $s(t)$ is translated in frequency to the right by ω_c (assumed to be positive), as is verified by

$$\int_{-\infty}^{\infty} s(t)e^{i\omega_c t}e^{-i\omega t}d\omega = \int_{-\infty}^{\infty} s(t)e^{-i(\omega-\omega_c)t}d\omega = S(\omega - \omega_c) \tag{6.2.5}$$

This formula describes the essence of the so-called *complex modulation process*. (An analogous formula for random discrete-time signals is given by equation (1.4.11) in Chapter 1.) The output of the complex modulation process is always complex valued (hence the name of this form of modulation). If the modulated signal is real valued, as $x(t)$ is, then it must have an even spectrum. In such a case, the translation of $S(\omega)$ to the right by ω_c, as in (6.2.5), must be accompanied by a translation to the left (also by ω_c) of the folded and complex-conjugated baseband spectrum. This process results in the following expression for $X(\omega)$:

$$\boxed{X(\omega) = S(\omega - \omega_c) + S^*(-(\omega + \omega_c)\ )} \tag{6.2.6}$$

It is readily verified that, in the time domain, the *real modulation process* leading to (6.2.6) corresponds to taking the real part of the complex-modulated signal $s(t)e^{i\omega_c t}$; that is,

$$\begin{aligned} x(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}[S(\omega - \omega_c) + S^*(-\omega - \omega_c)]e^{i\omega t}d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega - \omega_c)e^{i(\omega-\omega_c)t}e^{i\omega_c t}d\omega \\ &\qquad + \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} S(-\omega - \omega_c)e^{-i(\omega+\omega_c)t}e^{i\omega_c t}d\omega\right]^* \\ &= s(t)e^{i\omega_c t} + [s(t)e^{i\omega_c t}]^* \end{aligned}$$

which gives

$$\boxed{x(t) = 2\mathrm{Re}[s(t)e^{i\omega_c t}]} \tag{6.2.7}$$

or

$$\boxed{x(t) = 2\alpha(t)\cos(\omega_c t + \varphi(t))} \tag{6.2.8}$$

where $\alpha(t)$ and $\varphi(t)$ are the amplitude and phase of $s(t)$, respectively:

$$s(t) = \alpha(t)e^{i\varphi(t)}$$

If we let $s_I(t)$ and $s_Q(t)$ denote the real and imaginary parts of $s(t)$, then we can also write (6.2.7) as

$$\boxed{x(t) = 2[s_I(t)\cos(\omega_c t) - s_Q(t)\sin(\omega_c t)]} \tag{6.2.9}$$

We note in passing the following terminology associated with the equivalent time-domain representations (6.2.7)–(6.2.9) of a bandpass signal: $s(t)$ is called *the complex envelope* of $x(t)$, and $s_I(t)$ and $s_Q(t)$ are said to be *the in-phase and quadrature components* of $x(t)$.

The Demodulation Process. A calculation similar to (6.2.5) shows that the Fourier transform of $x(t)e^{-i\omega_c t}$ is given by

$$[S(\omega) + S^*(-\omega - 2\omega_c)]$$

which is simply $X(\omega)$ translated in frequency to the left by ω_c. The baseband (or lowpass) signal $s(t)$ can then be obtained by filtering $x(t)e^{-i\omega_c t}$ with a *baseband (or lowpass) filter* whose bandwidth is matched to that of $S(\omega)$. The hardware implementation of the demodulation process in block diagram form is presented in Figure 6.4.

6.2.2 Derivation of the Model Equation

Given the background of the previous subsection, we return to equation (6.2.4) describing the output of sensor k. Since $x(t)$ is assumed to be a bandpass signal, $X(\omega)$ is given by (6.2.6) which, when inserted in (6.2.4), leads to

$$\bar{Y}_k(\omega) = \bar{H}_k(\omega)[S(\omega - \omega_c) + S^*(-\omega - \omega_c)]e^{-i\omega\tau_k} + \bar{E}_k(\omega) \tag{6.2.10}$$

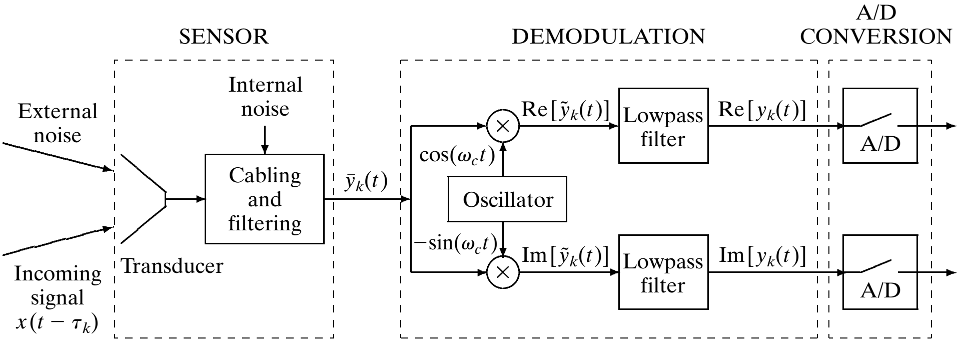


Figure 6.4 A simplified block diagram of the analog processing in a receiving-array element.

Let $\tilde{y}_k(t)$ denote the demodulated signal:

$$\tilde{y}_k(t) = \bar{y}_k(t)e^{-i\omega_c t}$$

It follows from (6.2.10) and the previous discussion on the demodulation process that the Fourier transform of $\tilde{y}_k(t)$ is given by

$$\begin{aligned}\tilde{Y}_k(\omega) = {} & \bar{H}_k(\omega+\omega_c)[S(\omega) + S^*(-\omega-2\omega_c)]e^{-i(\omega+\omega_c)\tau_k} \\ & + \bar{E}_k(\omega+\omega_c)\end{aligned} \tag{6.2.11}$$

When $\tilde{y}_k(t)$ is passed through a lowpass filter with bandwidth matched to $S(\omega)$, in the filter output (say, $y_k(t)$) the component in (6.2.11) centered at $\omega = -2\omega_c$ is eliminated along with all the other frequency components that fall in the stopband of the lowpass filter. Hence, we obtain

$$Y_k(\omega) = H_k(\omega+\omega_c)S(\omega)e^{-i(\omega+\omega_c)\tau_k} + E_k(\omega+\omega_c) \tag{6.2.12}$$

where $H_k(\omega+\omega_c)$ and $E_k(\omega+\omega_c)$ denote the parts of $\bar{H}_k(\omega+\omega_c)$ and $\bar{E}_k(\omega+\omega_c)$ that fall within the lowpass filter's passband, Ω, and where the frequency ω is restricted to Ω.

We now make the following *key assumption*:

The received signals are narrowband, so that $\lvert S(\omega)\rvert$ decreases rapidly with increasing $\lvert\omega\rvert$.	(6.2.13)

Under this assumption, (6.2.12) reduces (in an approximate way) to the following equation:

$$Y_k(\omega) = H_k(\omega_c)S(\omega)e^{-i\omega_c\tau_k} + E_k(\omega+\omega_c) \qquad \text{for } \omega \in \Omega \tag{6.2.14}$$

Because $H_k(\omega_c)$ must be different from zero, the sensor transfer function $\bar{H}_k(\omega)$ should pass frequencies near $\omega = \omega_c$ (as expected, since ω_c is the center frequency of the received signal). Also, note that we do not replace $E_k(\omega+\omega_c)$ in (6.2.14) by $E_k(\omega_c)$, as this term might not be (nearly) constant over the signal bandwidth. (For instance, this would be the case when the noise term in (6.2.12) contains a narrowband interference with the same center frequency as the signal.)

Remark: It is sometimes claimed that (6.2.12) can be reduced to (6.2.14) even if the *signals are broadband*, but *the sensors in the array are narrowband* with center frequency $\omega = \omega_c$. Under such an assumption, $\lvert H_k(\omega+\omega_c)\rvert$ goes quickly to zero as $\lvert\omega\rvert$ increases; hence, (6.2.12) becomes

$$Y_k(\omega) = H_k(\omega+\omega_c)S(0)e^{-i\omega_c\tau_k} + E_k(\omega+\omega_c) \tag{6.2.15}$$

which apparently is different from (6.2.14). In order to obtain (6.2.14) from (6.2.12) under the previous conditions, we need to make some additional assumptions. Hence, if we further assume that *the sensor frequency response is flat over the passband* (so that $H_k(\omega+\omega_c) = H_k(\omega_c)$) and that *the signal spectrum varies over the sensor passband* (so that $S(\omega)$ differs quite a bit from $S(0)$ over the passband in question), then we can still obtain (6.2.14) from (6.2.12). ■

The model of the array is derived in a straightforward manner from equation (6.2.14). The time-domain counterpart of (6.2.14) is the following:

$$y_k(t) = H_k(\omega_c)e^{-i\omega_c\tau_k}s(t) + e_k(t) \tag{6.2.16}$$

where $y_k(t)$ and $e_k(t)$ are the inverse Fourier transforms of the corresponding terms in (6.2.14). (By a slight abuse of notation, $e_k(t)$ is associated with $E_k(\omega+\omega_c)$, not $E_k(\omega)$.)

The hardware implementation required to obtain $\{y_k(t)\}$, as defined above, is indicated in Figure 6.4. Note that the scheme in Figure 6.4 generates samples of the real and imaginary components of $y_k(t)$. These samples are paired in the digital machine following the analog scheme of Figure 6.4 to obtain samples of the complex-valued signal $y_k(t)$. (We stress once more that all physical analog signals are real valued.) Note that the continuous-time signal in (6.2.16) is *bandlimited*: According to (6.2.14) (and the related discussion), $Y_k(\omega)$ is approximately equal to zero for $\omega \notin \Omega$. Here Ω is the support of $S(\omega)$ (recall that the filter bandwidth is matched to the signal bandwidth); hence, it is a narrow interval. Consequently, we can sample (6.2.16) with a rather low sampling frequency.

The sampled version of $\{y_k(t)\}$ is used by the "digital processing equipment" for the purpose of DOA estimation. Of course, the *digital form* of $\{y_k(t)\}$ satisfies an equation directly analogous to (6.2.16). In fact, to avoid a complication of notation by the introduction of a new discrete-time variable, *from here on we consider that t in equation (6.2.16) takes discrete values*:

$$t = 1, 2, \ldots, N \tag{6.2.17}$$

(As usual, we choose the sampling period as the unit of the time axis.) In Figure 6.4 we sample the baseband signal, which may be done by using sampling rates lower than those needed for the bandpass signal. (See also [Proakis, Rader, Ling, and Nikias 1992].)

Next, we introduce the so-called *array transfer vector* (or *direction vector*):

$$\boxed{a(\theta) = \left[H_1(\omega_c)e^{-i\omega_c\tau_1} \ldots H_m(\omega_c)e^{-i\omega_c\tau_m}\right]^T} \tag{6.2.18}$$

Here, θ denotes the *source's direction of arrival*, which is the parameter of interest in our problem. Note that, since the transfer characteristics and positions of the sensors in the array are assumed to be known, the vector in (6.2.18) is a function of θ only, as indicated by notation (this fact will be illustrated shortly by means of a particular form of array). By making use of (6.2.18), we can write equation (6.2.16) as

$$y(t) = a(\theta)s(t) + e(t) \tag{6.2.19}$$

where

$$y(t) = [y_1(t) \ldots y_m(t)]^T$$
$$e(t) = [e_1(t) \ldots e_m(t)]^T$$

denote the *array's output vector* and the *additive noise vector*, respectively. It should be noted that θ enters in (6.2.18) not only through $\{\tau_k\}$ but also through $\{H_k(\omega_c)\}$. In some cases, the sensors may be considered to be *omnidirectional* over the DOA range of interest, and then the $\{H_k(\omega_c)\}_{k=1}^m$ are independent of θ. Sometimes, the sensors may also be assumed to be *identical*. Then, by *redefining the signal* ($H(\omega_c)s(t)$ is redefined as $s(t)$) and *selecting the first sensor as the reference point*, the expression (6.2.18) can be simplified to the following form:

$$\boxed{a(\theta) = [1 \quad e^{-i\omega_c\tau_2} \ldots e^{-i\omega_c\tau_m}]^T} \tag{6.2.20}$$

The extension of equation (6.2.19) to the case of *multiple sources* is straightforward. Since the sensors in the array were assumed to be linear elements, a direct application of the *superposition principle* leads to the following *model of the array*:

$$\boxed{\begin{aligned} y(t) &= [a(\theta_1) \ldots a(\theta_n)] \begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix} + e(t) \triangleq As(t) + e(t) \\ \theta_k &= \text{the DOA of the } k\text{th source} \\ s_k(t) &= \text{the signal corresponding to the } k\text{th source} \end{aligned}} \tag{6.2.21}$$

It is interesting to note that the previous model equation mainly relies on the *narrowband assumption* (6.2.13). The *planar wave* assumption made in the introductory part of this chapter has *not* been used so far. *This assumption is to be used when deriving the explicit dependence of* $\{\tau_k\}$ *as a function of* θ, as is illustrated next for an array with a special geometry.

Uniform Linear Array. Consider the array of m identical sensors uniformly spaced on a line, depicted in Figure 6.5. Such an array is commonly referred to as a uniform linear array (ULA). Let d denote the distance between two consecutive sensors, and let θ denote the DOA of the signal illuminating the array, as measured (counterclockwise) with respect to the normal to the line of sensors. Then, under the planar wave hypothesis and the assumption that the first sensor in the array is chosen as the reference point, we find that

$$\tau_k = (k-1)\frac{d \sin\theta}{c} \qquad \text{for } \theta \in [-90^\circ, 90^\circ] \tag{6.2.22}$$

where c is the propagation velocity of the impinging waveform (for example, the speed of light, in the case of electromagnetic waves). Inserting (6.2.22) into (6.2.20) gives

$$a(\theta) = \left[1, e^{-i\omega_c d \sin\theta/c}, \ldots, e^{-i(m-1)\omega_c d \sin\theta/c}\right]^T \tag{6.2.23}$$

The restriction of θ to lie in the interval $[-90^\circ, 90^\circ]$ is a limitation of ULAs: two sources at locations symmetric with respect to the array line yield identical sets of delays $\{\tau_k\}$ and hence

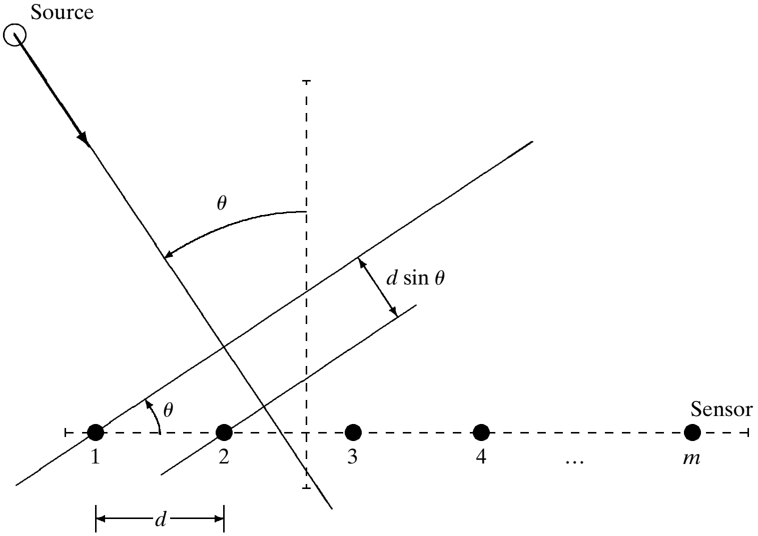


Figure 6.5 The uniform linear array scenario.

cannot be distinguished from one another. In practice, this ambiguity of ULAs is eliminated by using sensors that pass only signals whose DOAs are in $[-90^\circ, 90^\circ]$.

Let λ denote the *signal wavelength* (which is the distance traveled by the waveform in one period of the carrier):

$$\lambda = c/f_c, \qquad f_c = \omega_c/2\pi \tag{6.2.24}$$

Define

$$f_s = f_c \frac{d \sin\theta}{c} = \frac{d \sin\theta}{\lambda} \tag{6.2.25}$$

and

$$\boxed{\omega_s = 2\pi f_s = \omega_c \frac{d \sin\theta}{c}} \tag{6.2.26}$$

With this notation, the transfer vector (6.2.23) can be rewritten as

$$\boxed{a(\theta) = \left[1 \;\; e^{-i\omega_s} \ldots e^{-i(m-1)\omega_s}\right]^T} \tag{6.2.27}$$

This is a Vandermonde vector that is completely analogous to the vector made from the uniform samples of the sinusoidal signal $\{e^{-i\omega_s t}\}$. Let us explore this analogy a bit further.

First, by the previous analogy, ω_s is called the *spatial frequency*.

Second, if we were to sample a continuous-time sinusoidal signal with frequency ω_c, then, in order to avoid aliasing effects, the sampling frequency f_0 should satisfy (by the Nyquist sampling theorem)

$$f_0 > 2f_c \tag{6.2.28}$$

or, equivalently,

$$T_0 < \frac{T_c}{2} \tag{6.2.29}$$

where T_0 is the sampling period and T_c is the period of the continuous-time sinusoidal signal. Now, in the ULA case considered in this example, we see from (6.2.27) that the vector $a(\theta)$ is uniquely defined (i.e., there is no "spatial aliasing") if and only if ω_s is constrained as follows:

$$|\omega_s| < \pi \tag{6.2.30}$$

However, (6.2.30) is equivalent to

$$|f_s| < \frac{1}{2} \Longleftrightarrow d|\sin\theta| < \frac{\lambda}{2} \tag{6.2.31}$$

Note that this condition on d depends on θ. In particular, for a broadside source (i.e., a source with $\theta = 0°$), (6.2.31) imposes *no* constraint on d. However, in general, we have no knowledge about the DOA of the source signal. Consequently, we would like (6.2.31) to hold for *any* θ, which leads to the following condition on d:

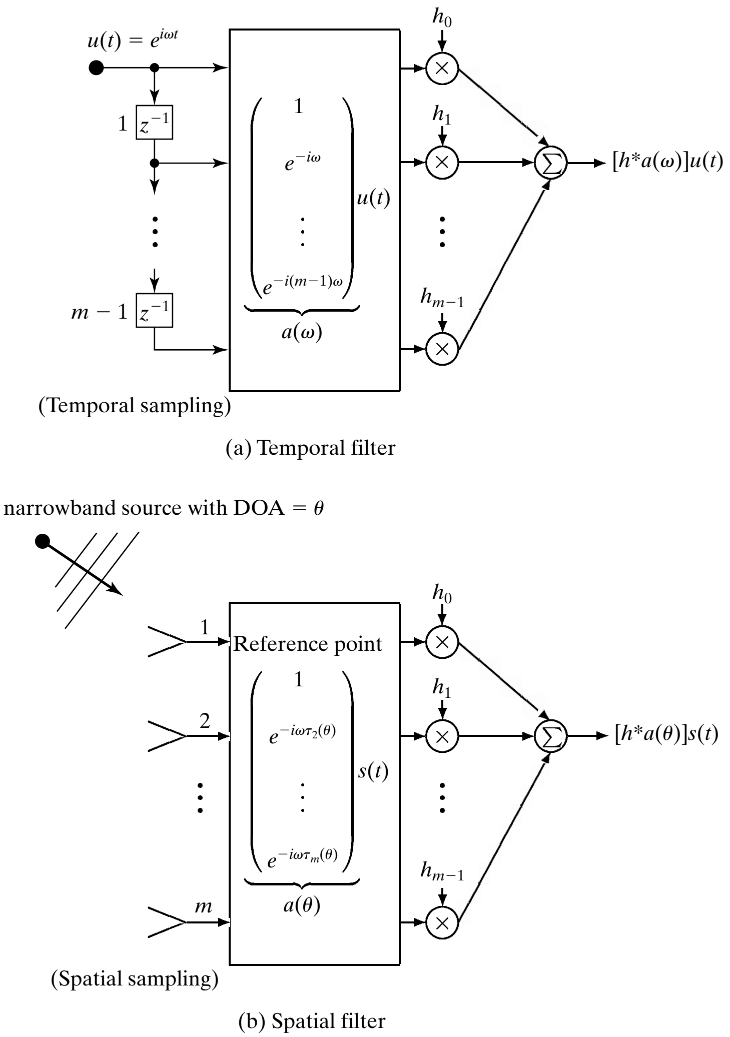
$$\boxed{d < \frac{\lambda}{2}} \tag{6.2.32}$$

We may think of the ULA as performing a uniform spatial sampling of the wavefield, so equation (6.2.32) simply says that the (spatial) sampling period d should be smaller than half of the signal wavelength. By analogy with (6.2.29), this result may be interpreted as a *spatial Nyquist sampling theorem*.

Equipped with the array model (6.2.21) derived previously, we can reduce the problem of DOA finding to that of estimating the parameters $\{\theta_k\}$ in (6.2.21). As there is a *direct analogy between (6.2.21) and the model (4.2.6) for sinusoidal signals in noise*, we may expect that most of the methods developed in Chapter 4 for (temporal) frequency estimation can also be used for DOA estimation. This is shown to be the case in the next sections, which briefly review the most important DOA finding methods.

6.3 NONPARAMETRIC METHODS

The methods to be described in this section *do not make any assumption about the covariance structure of the data*. As such, they may be considered to be "nonparametric." On the other hand, they assume that *the functional form of the array's transfer vector $a(\theta)$ is known*. Can we then still categorize them as "nonparametric methods"? The array performs a spatial sampling of the



(a) Temporal filter

(b) Spatial filter

Figure 6.6 Analogy between temporal sampling and filtering and the corresponding spatial operations performed by an array of sensors.

incoming wavefront, which is analogous to the temporal sampling done by the tapped-delay line implementation of a (temporal) finite impulse response (FIR) filter; see Figure 6.6. Thus, assuming that the form of $a(\theta)$ is available is no more restrictive than making the same assumption for $a(\omega)$ in Figure 6.6(a). In conclusion, the functional form of $a(\theta)$ characterizes the array as a *spatial sampling device* and, assuming it is known, should not be considered to be parametric (or model-based) information. As already mentioned, an array for which the functional form of $a(\theta)$ is known is said to be *calibrated*.

Figure 6.6 also makes an analogy between *temporal FIR filtering* and *spatial filtering* using an array of sensors. In what follows, we comment briefly on this analogy, since it is of interest for the nonparametric approach to DOA finding. In the time-series case, a FIR filter is defined by the relation

$$y_F(t) = \sum_{k=0}^{m-1} h_k u(t-k) \triangleq h^* y(t) \tag{6.3.1}$$

where $\{h_k\}$ are the filter weights, $u(t)$ is the input to the filter, and

$$h = [h_0 \ldots h_{m-1}]^* \tag{6.3.2}$$

$$y(t) = [u(t) \ldots u(t-m+1)]^T \tag{6.3.3}$$

Similarly, we can use the spatial samples $\{y_k(t)\}_{k=1}^{m}$ obtained with a sensor array to define a *spatial filter*:

$$\boxed{y_F(t) = h^* y(t)} \tag{6.3.4}$$

A temporal filter can be made to enhance or attenuate some selected frequency bands by choosing the vector h appropriately. More precisely, since the filter output for a sinusoidal input $u(t)$ is given by

$$y_F(t) = [h^* a(\omega)] u(t) \tag{6.3.5}$$

(where $a(\omega)$ is as defined, for instance, in Figure 6.6), then, by selecting h so that $h^* a(\omega)$ is large (small), we can enhance (attenuate) the power of $y_F(t)$ at frequency ω.

In direct analogy with (6.3.5), the (noise-free) spatially filtered output (as in (6.3.4)) of an array illuminated by a narrowband wavefront with complex envelope $s(t)$ and DOA equal to θ is given by (*cf.* (6.2.19)):

$$\boxed{y_F(t) = [h^* a(\theta)] s(t)} \tag{6.3.6}$$

This equation clearly shows that *the spatial filter can be selected to enhance (attenuate) the signals coming from a given direction* θ, by making $h^* a(\theta)$ in (6.3.6) large (small). This observation lies at the basis of the DOA-finding methods to be described in this section. All of these methods can be derived by using the *filter-bank approach* of Chapter 5. More specifically, assume that a filter h has been found such that

(i) it passes undistorted the signals with a given DOA θ; and
(ii) it attenuates all the other DOAs different from θ as much as possible. (6.3.7)

Then the power of the spatially filtered signal in (6.3.4),

$$\boxed{E\left\{|y_F(t)|^2\right\} = h^*Rh, \qquad R = E\left\{y(t)y^*(t)\right\}} \tag{6.3.8}$$

should give a good indication of the energy coming from direction θ. (Note that θ enters in (6.3.8) via h.) Hence, *h^*Rh should peak at the DOAs of the sources located in the array's viewing field* when evaluated over the DOA range of interest. This fact may be exploited for the purpose of DOA finding. Depending on the specific way in which the (loose) design objectives in (6.3.7) are formulated, the above approach can lead to different DOA estimation methods. In the following, we present *spatial extensions of the periodogram and Capon techniques*. The *RFB method* of Chapter 5 may also be extended to the spatial processing case, provided the array's geometry is such that the transfer vector $a(\theta + \alpha)$ can be factored as

$$a(\theta + \alpha) = D(\theta)a(\alpha) \tag{6.3.9}$$

where D is a unitary (possibly diagonal) matrix. Without such a property, the RFB spatial filter should be computed, *for each θ*, by solving an $m \times m$ eigendecomposition problem, which would be computationally prohibitive in most applications. It is not *a priori* obvious that an arbitrary array satisfies (6.3.9), so we do not consider the RFB approach in what follows.[1] Finally, we remark that a spatial filter satisfying the design objectives in (6.3.7) can be viewed as *forming a (reception) beam* in the direction θ, as pictorially indicated in Figure 6.7. Because of this interpretation, the methods resulting from this approach to the DOA-finding problem, in particular the method of the next subsection, are called *beamforming methods* [Van Veen and Buckley 1988; Johnson and Dudgeon 1992].

6.3.1 Beamforming

In view of (6.3.6), *condition* (i) of the filter-design problem (6.3.7) can be formulated as

$$h^*a(\theta) = 1 \tag{6.3.10}$$

In what follows, we assume that the transfer vector $a(\theta)$ has been normalized so that

$$a^*(\theta)a(\theta) = m \tag{6.3.11}$$

Note that, in the case of an array with identical sensors, the condition (6.3.11) is automatically met (*cf.* (6.2.20)).

Regarding *condition* (ii) in (6.3.7), if $y(t)$ in (6.3.8) were *spatially white with $R = I$*, then we would obtain the following expression for the power of the filtered signal:

$$E\left\{|y_F(t)|^2\right\} = h^*h \tag{6.3.12}$$

[1] Referring back to Chapter 5 may prove useful for understanding these comments on RFB and for several other discussions in this section.

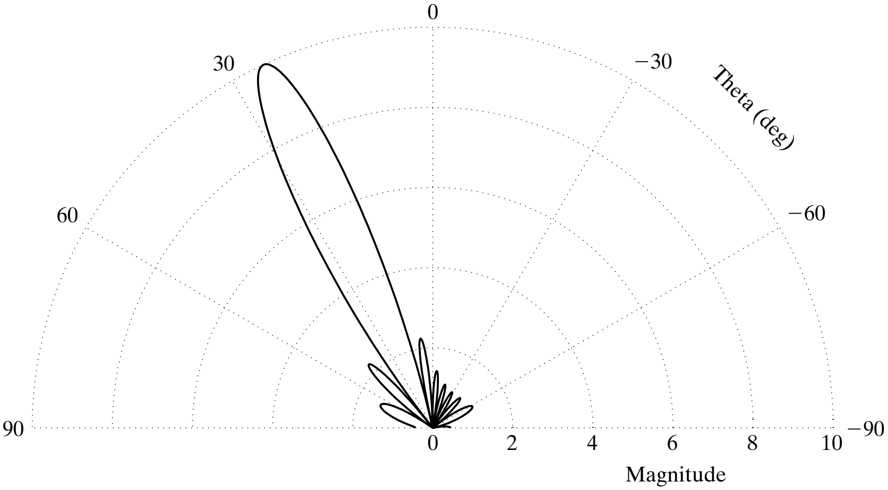


Figure 6.7 The response magnitude $|h^* a(\theta)|$, versus θ, of a spatial filter (or beamformer). Here, $h = a(\theta_0)$, where $\theta_0 = 25^\circ$ is the DOA of interest; the array is a 10-element ULA with $d = \lambda/2$.

This is different from zero for every θ (note that we cannot have $h = 0$, because of condition (6.3.10)), a fact which indicates that *a spatially white signal in the array output can be considered as having equal power for all directions* θ (in the same manner as a temporally white signal contains equal power in all frequency bands). We deduce from this observation that a natural mathematical formulation of condition (ii) would be to require that h minimize the power in (6.3.12). Hence, we are led to the following design problem:

$$\boxed{\min_h h^* h \quad \text{subject to} \quad h^* a(\theta) = 1} \tag{6.3.13}$$

Because (6.3.13) is a special case of the optimization problem (5.4.7) in Chapter 5, we obtain the solution to (6.3.13) from (5.4.8) as

$$h = a(\theta)/a^*(\theta)a(\theta) \tag{6.3.14}$$

Making use of (6.3.11) reduces (6.3.14) to

$$\boxed{h = a(\theta)/m} \tag{6.3.15}$$

which, when inserted in (6.3.8), gives

$$E\left\{|y_F(t)|^2\right\} = a^*(\theta) R a(\theta)/m^2 \tag{6.3.16}$$

The theoretical covariance matrix R in (6.3.16) cannot be (exactly) determined from the available finite sample $\{y(t)\}_{t=1}^{N}$; hence, it must be replaced by some estimate, such as

$$\boxed{\hat{R} = \frac{1}{N}\sum_{t=1}^{N} y(t)y^*(t)} \tag{6.3.17}$$

By doing so and omitting the factor $1/m^2$ in (6.3.16), which has no influence on the DOA estimates, we obtain the *beamforming method*, which estimates the DOAs as summarized in the next box.

$$\boxed{\begin{array}{l}\text{The beamforming DOA estimates are given by the locations of the } n \text{ highest}\\ \text{peaks of the function}\\ \qquad\qquad\qquad a^*(\theta)\hat{R}a(\theta)\end{array}} \tag{6.3.18}$$

When the estimated spatial spectrum in (6.3.18) is compared to the expression derived in Section 5.4 for the Blackman–Tukey periodogram, it is seen that *beamforming is a direct (spatial) extension of the periodogram*. In fact, the function in (6.3.18) may be thought of as being obtained by averaging the "spatial periodograms"

$$|a^*(\theta)y(t)|^2 \tag{6.3.19}$$

over the set of available "snapshots" ($t = 1, \ldots, N$).

The connection established in the previous paragraph, between beamforming and the (averaged) periodogram, suggests that the *resolution properties* of the beamforming method are analogous to those of the periodogram method. In fact, by an analysis similar to that in Chapters 2 and 5, it can be shown that the *beamwidth*[2] of the spatial filter used by beamforming is approximately equal to the inverse of the array's aperture (as measured in signal wavelengths). This sets a limit on the resolution achievable with beamforming (see Exercise 6.2):

$$\boxed{\text{Beamforming DOA resolution limit} \simeq \text{wavelength / array “length”}} \tag{6.3.20}$$

Next, we note that, as N increases, the sample spatial spectrum in (6.3.18) converges (under mild conditions) to (6.3.16), uniformly in θ. Hence, the beamforming estimates of the DOAs converge to the n maximum points of (6.3.16), as N tends to infinity. *If the array model (6.2.21) holds* (it has not been used so far!), *the noise $e(t)$ is spatially white and has the same power σ^2 in all sensors*, and *if there is only one source* (with DOA denoted by θ_0, for convenience), then R in (6.3.16) is given by

$$R = a(\theta_0)a^*(\theta_0)P + \sigma^2 I \tag{6.3.21}$$

[2] The beamwidth is the spatial counterpart of the temporal notion of bandwidth associated with a bandpass filter.

where $P = E\left\{|s(t)|^2\right\}$ denotes the signal power. Hence,

$$\begin{aligned} a^*(\theta)Ra(\theta) &= |a^*(\theta)a(\theta_0)|^2 P + a^*(\theta)a(\theta)\sigma^2 \\ &\leq |a^*(\theta)a(\theta)||a^*(\theta_0)a(\theta_0)|P + \sigma^2 a^*(\theta)a(\theta) \\ &= m(mP + \sigma^2) \end{aligned} \tag{6.3.22}$$

where the inequality follows from the Cauchy–Schwartz lemma (see Result R22 in Appendix A) and the last equality follows from (6.3.11). The upper bound in (6.3.22) is achieved for $a(\theta) = a(\theta_0)$, which, under mild conditions, implies $\theta = \theta_0$. In conclusion, the *beamforming DOA estimate is consistent under the previous assumptions* ($n = 1$, *etc.*). *In the general case of multiple sources, however, the DOA estimates obtained with beamforming are inconsistent.* The (asymptotic) bias of these estimates may be significant if the sources are strongly correlated or closely spaced.

As was explained before, beamforming is the spatial analog of the Blackman–Tukey periodogram (with a certain covariance estimate) and of the Bartlett periodogram (if we interpret the m-dimensional snapshots in (6.3.19) as "subsamples" of the available "sample" $[y^T(1), \ldots, y^T(N)]^T$). Note, however, that the value of m in the periodogram methods can be chosen by the user, whereas, in the beamforming method, m is fixed. This difference might seem small at first, but it has a significant impact on the consistency properties of beamforming. More precisely, it can be shown that, for instance, the Bartlett periodogram estimates of *temporal frequencies* are *consistent* under the model (4.2.7), *provided that* m increases without bound as the number of samples N tends to infinity (e.g., we can set $m = N$, which yields the unmodified periodogram).[3] For beamforming, on the other hand, the value of m (i.e., the number of array elements) is *limited* by physical considerations. This prevents beamforming from providing consistent DOA estimates in the multiple-signal case. An additional difficulty is that, in the spatial scenario, the signals can be correlated with one another, whereas they are always uncorrelated in the temporal frequency estimation case. Explaining why this is so and completing a consistency analysis of the beamforming DOA estimates is left as an exercise for the reader.

Now, if the model (6.2.21) holds, if the minimum DOA separation is larger than the array beamwidth (which implies that m is sufficiently large), if the signals are uncorrelated, and if the noise is spatially white, then it is readily seen that the multiple-source spectrum (6.3.16) decouples (approximately) into n single-source spectra; this means that beamforming can provide reasonably accurate DOA estimates in such a case. In fact, in this case, beamforming can be shown to provide an approximation to the nonlinear LS DOA estimation method discussed in Section 6.4.1; see the remark in that section.

6.3.2 Capon Method

The derivation of the Capon method for array signal processing is entirely analogous to the derivation of the Capon method for time series data developed in Section 5.4 [Capon 1969; Lacoss

[3] The unmodified periodogram is an *inconsistent estimator* for *continuous* PSDs (as shown in Chapter 2). However, as already asserted, the plain periodogram estimates of *discrete (or line)* PSDs are *consistent*. Showing this is left as an exercise to the reader. (Make use of the covariance matrix model (4.2.7) with $m \to \infty$ and of the fact that the Fourier (or Vandermonde) vectors, at different frequencies, become orthogonal to one another as their dimension increases.)

1971]. The Capon spatial filter design problem is the following:

$$\boxed{\min_h h^* R h \quad \text{subject to} \quad h^* a(\theta) = 1} \tag{6.3.23}$$

Hence, objective (i) in the general design problem (6.3.7) is ensured by constraining the filter exactly as in the beamforming approach. (See (6.3.10).) Objective (ii) in (6.3.7) is accomplished by requiring the filter to minimize the output power, when fed with the actual array data $\{y(t)\}$. Hence, in the Capon approach, objective (ii) is formulated in a "data-dependent" way, whereas it is formulated independently of the data in the beamforming method. As a consequence, the goal of the Capon filter steered to a certain direction θ is to attenuate any other signal that *actually impinges on the array* from a DOA $\neq \theta$, whereas the beamforming filter pays uniform attention to *all other* DOAs $\neq \theta$, even though there might be no incoming signal for many of those DOAs.

The solution to (6.3.23), as derived in Section 5.4, is given by

$$\boxed{h = \frac{R^{-1} a(\theta)}{a^*(\theta) R^{-1} a(\theta)}} \tag{6.3.24}$$

which, when inserted in the output power formula (6.3.8), leads to

$$E\left\{|y_F(t)|^2\right\} = \frac{1}{a^*(\theta) R^{-1} a(\theta)} \tag{6.3.25}$$

All that remains is to replace R in (6.3.25) by a sample estimate, such as $\hat{R}$ in (6.3.17), to obtain the Capon DOA estimator.

$$\boxed{\begin{array}{l}\text{The Capon DOA estimates are obtained as the locations of the } n \text{ largest peaks} \\ \text{of the following function:} \\ \qquad\qquad\qquad\qquad \dfrac{1}{a^*(\theta)\hat{R}^{-1}a(\theta)}\end{array}} \tag{6.3.26}$$

There is an implicit assumption in (6.3.26) that $\hat{R}^{-1}$ exists, but this can be ensured under weak conditions (in particular, $\hat{R}^{-1}$ exists with probability 1 if $N \geq m$ and if the noise term has a positive definite spatial covariance matrix). Note that the "spatial spectrum" in (6.3.26) corresponds to the "CM-Version 1" PSD for time series. (See equation (5.4.12) in Section 5.4.) A Capon spatial spectrum similar to the "CM-Version 2" PSD formula (see (5.4.17)) might also be derived, but it appears to be more complicated than the time series formula if the array is not a ULA.

Capon DOA estimation has been found empirically to possess superior performance as compared with beamforming. The common advantage of these two nonparametric methods is that they do not assume anything about the statistical properties of the data, and, therefore, they can be used in situations where we lack information about these properties. On the other hand, in the cases where such information is available, for example in the form of a covariance model of the data, a nonparametric approach does not give the performance that one can achieve with a parametric (model-based) approach. The parametric approach to DOA estimation is the subject of the next section.

6.4 PARAMETRIC METHODS

In this section, *we postulate the array model (6.2.21)*. Furthermore, the noise $e(t)$ is assumed to be spatially white with components having identical variance:

$$\boxed{E\left\{e(t)e^*(t)\right\} = \sigma^2 I} \tag{6.4.1}$$

In addition, the signal covariance matrix

$$P = E\left\{s(t)s^*(t)\right\} \tag{6.4.2}$$

is assumed to be *nonsingular* (but not necessarily diagonal; hence, the signals may be (partially) correlated). When the signals are fully correlated, so that P is singular, they are said to be *coherent*. Finally, we assume that the signals and the noise are uncorrelated with one another.

Under the previous assumptions, the theoretical covariance matrix of the array output vector is given by

$$\boxed{R = E\left\{y(t)y^*(t)\right\} = APA^* + \sigma^2 I} \tag{6.4.3}$$

There is a direct analogy between the array models (6.2.21) and (6.4.3) and the corresponding models encountered in our discussion of the sinusoids-in-noise case in Chapter 4. More specifically, the "nonlinear regression" model (6.2.21) of the array is analogous to (4.2.6), and the array covariance model (6.4.3) is much the same as (4.2.7). The consequence of these analogies is that *all methods introduced in Chapter 4 for frequency estimation can also be used for DOA estimation* without any essential modification. In the following, we briefly review these methods, with a view to pointing out any differences from the frequency-estimation application. When the assumed array model is a good representation of reality, the parametric DOA estimation methods provide highly accurate DOA estimates, even in adverse situations (such as low SNR scenarios). Our main thrust in this text has been to understand the basic ideas behind the presented spectral estimation methodologies, so we do not dwell on the details of the analysis required to establish the statistical properties of the DOA estimators discussed in what follows; see, however, Appendix B for a discussion on the Cramér–Rao bound and the best accuracy achievable in DOA estimation problems. Such analysis details are available in [Stoica and Nehorai 1989a; Stoica and Nehorai 1990; Stoica and Sharman 1990; Stoica and Nehorai 1991; Viberg and Ottersten 1991; Rao and Hari 1993]. For reviews of many of the recent advances in spatial-spectral analysis, the reader can consult [Pillai 1989], [Ottersten, Viberg, Stoica, and Nehorai 1993], and [Van Trees 2002].

6.4.1 Nonlinear Least-Squares Method

This method finds the unknown DOAs as the minimizing elements of the following function:

$$f = \frac{1}{N}\sum_{t=1}^{N} \| y(t) - As(t) \|^2 \tag{6.4.4}$$

Minimization with respect to $\{s(t)\}$ (see Result R32 in Appendix A) gives

$$s(t) = (A^*A)^{-1}A^*y(t) \qquad t = 1, \dots, N \tag{6.4.5}$$

By inserting (6.4.5) into (6.4.4), we get the following concentrated nonlinear least-squares (LS) criterion:

$$\begin{aligned} f &= \frac{1}{N}\sum_{t=1}^{N} \| \{I - A(A^*A)^{-1}A^*\}y(t) \|^2 \\ &= \frac{1}{N}\sum_{t=1}^{N} y^*(t)\left[I - A(A^*A)^{-1}A^*\right]y(t) \\ &= \operatorname{tr}\{[I - A(A^*A)^{-1}A^*]\hat{R}\} \end{aligned} \tag{6.4.6}$$

The second equality in (6.4.6) follows from the fact that the matrix $I - A(A^*A)^{-1}A^*$ is idempotent (it is the orthogonal projector onto $\mathcal{N}(A^*)$) and the third equality follows from the properties of the trace operator. (See Result R8 in Appendix A.) From (6.4.6), the nonlinear LS DOA estimates are given by

$$\boxed{\{\hat{\theta}_k\} = \arg\max_{\{\theta_k\}} \operatorname{tr}\left[A(A^*A)^{-1}A^*\hat{R}\right]} \tag{6.4.7}$$

Remark: Much as in the frequency-estimation case, it can be shown that beamforming provides an approximate solution to the previous nonlinear LS problem whenever the DOAs are known to be well separated. To see this, let us assume that we restrict the search for the maximizers of (6.4.7) to a set of well-separated DOAs (according to the *a priori* information that the true DOAs belong to this set). In such a set, $A^*A \simeq mI$ under weak conditions; hence, the function in (6.4.7) can be approximately written as

$$\operatorname{tr}\left[A(A^*A)^{-1}A^*\hat{R}\right] \simeq \frac{1}{m}\sum_{k=1}^{n} a^*(\theta_k)\hat{R}a(\theta_k)$$

Paralleling the discussion following equation (4.3.16) in Chapter 4, we can show that the beamforming DOA estimates maximize the right-hand side of the previous equation over the set under consideration. With this observation, the proof of the fact that the computationally efficient beamforming method provides an approximate solution to (6.4.7) in scenarios with well-separated DOAs is concluded. ■

One difference between (6.4.7) and the corresponding optimization problem in the frequency-estimation application (see (4.3.8) in Section 4.3) lies in the fact that, in the frequency-estimation application, only one "snapshot" of data is available, in contrast to the N snapshots available in the DOA-estimation application. Another, more important difference is that, for non-ULA cases, the matrix A in (6.4.7) does not have the Vandermonde structure of the corresponding matrix in (4.3.8). As a consequence, several of the algorithms used to (approximately) solve the frequency-estimation problem (such as the one in [Kumaresan, Scharf, and Shaw 1986] and [Bresler and Macovski 1986]) are no longer applicable to solving (6.4.7) unless the array is a ULA.

6.4.2 Yule–Walker Method

The matrix Γ, which lies at the basis of the Yule–Walker method (see Section 4.4), can be constructed from any block of R in (6.4.3) that does not include diagonal elements. To be more precise, partition the array model (6.2.21) into the following two nonoverlapping parts:

$$y(t) = \begin{bmatrix} \bar{y}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} \bar{A} \\ \tilde{A} \end{bmatrix} s(t) + \begin{bmatrix} \bar{e}(t) \\ \tilde{e}(t) \end{bmatrix} \tag{6.4.8}$$

Because $\bar{e}(t)$ *and* $\tilde{e}(t)$ *are uncorrelated* (by assumption), we have

$$\boxed{\Gamma \triangleq E\left\{\bar{y}(t)\tilde{y}^*(t)\right\} = \bar{A}P\tilde{A}^*} \tag{6.4.9}$$

which is assumed to be of dimension $M \times L$ (with $M + L = m$). For

$$\boxed{M > n, \qquad L > n} \tag{6.4.10}$$

(which cannot hold unless $m > 2n$), the rank of Γ is equal to n (under weak conditions) and the $(L-n)$-dimensional null space of this matrix contains complete information about the DOAs. To see this, let G be an $L \times (L-n)$ matrix whose columns form a basis of $\mathcal{N}(\Gamma)$. (G can be obtained from the SVD of Γ; see Result R15 in Appendix A.) Then we have $\Gamma G = 0$, which implies (using the fact that $\text{rank}(\bar{A}P) = n$) that

$$\tilde{A}^* G = 0$$

This observation can be used, in the manner of Sections 4.4 (YW) and 4.5 (MUSIC), to estimate the DOAs from a sample estimate of Γ, such as

$$\hat{\Gamma} = \frac{1}{N}\sum_{t=1}^{N} \bar{y}(t)\tilde{y}^*(t) \tag{6.4.11}$$

Unlike all the other methods discussed in what follows, *the Yule–Walker method does not impose the rather stringent condition (6.4.1).* The Yule–Walker method requires only that $E\{\bar{e}(t)\tilde{e}^*(t)\} = 0$, which is a much weaker assumption. This is a distinct advantage of the Yule–Walker method (see [Viberg, Stoica, and Ottersten 1995] for details). Its relative drawback is that it cannot be used unless $m > 2n$ (all the other methods require only that $m > n$); in general, it has been found to provide accurate DOA estimates only in those applications involving large-aperture arrays.

Interestingly enough, whenever the condition (6.4.1) holds (i.e., the noise at the array output is spatially white), we can use a modification of the technique above that does not require that $m > 2n$ [Fuchs 1996]. To see this, let

$$\tilde{\Gamma} \triangleq E\left\{y(t)\tilde{y}^*(t)\right\} = R\begin{bmatrix} 0 \\ I_L \end{bmatrix} \qquad (m \times L)$$

where $\tilde{y}(t)$ is as defined in (6.4.8); hence $\tilde{\Gamma}$ is made from the last L columns of R. By making use of the expression (6.4.3) for R, we obtain

$$\tilde{\Gamma} = AP\tilde{A}^* + \sigma^2\begin{bmatrix} 0 \\ I_L \end{bmatrix} \tag{6.4.12}$$

Because the noise terms in $y(t)$ and $\tilde{y}(t)$ are correlated, the noise is still present in $\tilde{\Gamma}$ (as can be seen from (6.4.12)); hence, $\tilde{\Gamma}$ is not really a YW matrix. Nevertheless, $\tilde{\Gamma}$ has a property similar to that of the YW matrix Γ, as we now show.

First, observe that

$$\tilde{\Gamma}^*\tilde{\Gamma} = \tilde{A}(2\sigma^2 P + PA^*AP)\tilde{A}^* + \sigma^4 I$$

The matrix $2\sigma^2 P + PA^*AP$ is readily shown to be nonsingular if and only if P is nonsingular. As $\tilde{\Gamma}^*\tilde{\Gamma}$ has the same form as R in (6.4.3), we conclude that (for $m \geq L > n$) the $L \times (L-n)$ matrix $\tilde{G}$, whose columns are the eigenvectors of $\tilde{\Gamma}^*\tilde{\Gamma}$ that correspond to the multiple minimum eigenvalue of σ^4, satisfies

$$\tilde{A}^*\tilde{G} = 0 \tag{6.4.13}$$

The columns of $\tilde{G}$ are also equal to the $(L-n)$ right singular vectors of $\tilde{\Gamma}$ corresponding to the multiple minimum singular value of σ^2. For numerical precision reasons, $\tilde{G}$ should be computed from the singular vectors of $\tilde{\Gamma}$ rather than from the eigenvectors of $\tilde{\Gamma}^*\tilde{\Gamma}$ (see Section A.8.2).

Because (6.4.13) has the same form as $\tilde{A}^*G = 0$, we can use (6.4.13) for subspace-based DOA estimation in exactly the same way as we used $\tilde{A}^*G = 0$ (see equation (4.5.6) and the discussion following it in Chapter 4). Note that, for the method based on $\tilde{\Gamma}$ to be usable, we require only that

$$m \geq L > n \tag{6.4.14}$$

instead of the more restrictive conditions $\{m - L > n,\ L > n\}$ (see (6.4.10)) required in the YW method based on Γ. Observe that (6.4.14) can always be satisfied if $m > n$, whereas (6.4.10) requires that $m > 2n$. Finally, note that Γ is made from the first $m - L$ rows of $\tilde{\Gamma}$

and hence contains "less information" than $\tilde{\Gamma}$; this provides a quick intuitive explanation of why the method based on Γ requires more sensors to be applicable than does the method based on $\tilde{\Gamma}$.

6.4.3 Pisarenko and MUSIC Methods

The MUSIC algorithm (with Pisarenko as a special case), developed in Section 4.5 for the frequency-estimation application, can be used without modification for DOA estimation [Bienvenu 1979; Schmidt 1979; Barabell 1983]. There are only minor differences between the DOA and the frequency-estimation applications of MUSIC, as pointed out next.

First, in the spatial application, we can choose between the Spectral and Root MUSIC estimators only in the case of a ULA. For most of the other array geometries, *only Spectral MUSIC is applicable*.

Second, *the standard MUSIC algorithm (4.5.15) breaks down in the case of coherent signals*, because, in that case, the rank condition (4.5.1) no longer holds. (Such a situation cannot occur in the frequency-estimation application, because P is always (diagonal and) nonsingular there.) However, the *modified MUSIC algorithm (outlined at the end of Section 4.5) can be used when the signals are coherent, provided that the array is uniform and linear*. This is so because the property (4.5.23), on which the modified MUSIC algorithm is based, continues to hold even if P is singular. (See Exercise 6.14.)

6.4.4 Min–Norm Method

There is no essential difference between the use of the Min–Norm method for frequency estimation and for DOA-finding in the noncoherent case. As for MUSIC, in the DOA-estimation application, the Min–Norm method should not be used in scenarios with coherent signals, and the Root Min–Norm algorithm can only be used in the ULA case [Kumaresan and Tufts 1983]. In addition, the key property that the true DOAs are asymptotically the *unique* solutions of the Min–Norm estimation problem holds in the ULA case (see Complement 6.5.1), but not necessarily for other array geometries.

6.4.5 ESPRIT Method

In the ULA case, ESPRIT can be used for DOA estimation exactly as it is for frequency estimation. (See Section 4.7.) In the non-ULA case, ESPRIT can be used only in certain situations. More precisely, and unlike the other algorithms in this section, ESPRIT can be used for DOA finding only if *the array contains two identical subarrays that are displaced by a known displacement vector* [Roy and Kailath 1989; Stoica and Nehorai 1991]. Mathematically, this condition can be formulated as follows: Let $\bar{m}$ denote the number of sensors in the two twin subarrays, and let A_1 and A_2 denote the submatrices of A corresponding to these subarrays. The sensors in the array are numbered arbitrarily, so it constitutes no restriction to assume that A_1 is made from the first $\bar{m}$ rows in A and A_2 from the last $\bar{m}$:

$$A_1 = [I_{\bar{m}} \quad 0]A \qquad (\bar{m} \times n) \tag{6.4.15}$$

$$A_2 = [0 \quad I_{\bar{m}}]A \qquad (\bar{m} \times n) \tag{6.4.16}$$

Here $I_{\bar{m}}$ denotes the $\bar{m} \times \bar{m}$ identity matrix. Note that the two subarrays overlap if $\bar{m} > m/2$; otherwise, they might not overlap. If the array is purposely built to meet ESPRIT's subarray condition, then, normally, $\bar{m} = m/2$, and the two subarrays are nonoverlapping.

Mathematically, the ESPRIT requirement means that

$$\boxed{A_2 = A_1 D} \tag{6.4.17}$$

where

$$\boxed{D = \begin{bmatrix} e^{-i\omega_c \tau(\theta_1)} & & 0 \\ & \ddots & \\ 0 & & e^{-i\omega_c \tau(\theta_n)} \end{bmatrix}} \tag{6.4.18}$$

and where $\tau(\theta)$ denotes the time needed by a wavefront impinging upon the array from the direction θ to travel between (the "reference points" of) the two twin subarrays. If the angle of arrival θ is measured with respect to the perpendicular of the line between the subarrays' center points, then a calculation similar to the one that led to (6.2.22) shows that

$$\boxed{\tau(\theta) = d \sin(\theta)/c} \tag{6.4.19}$$

where d is the distance between the two subarrays. Hence, estimates of the DOAs can readily be derived from estimates of the diagonal elements of D in (6.4.18).

Equations (6.4.17) and (6.4.18) are basically equivalent to (4.7.3) and (4.7.4) in Section 4.7; hence, the ESPRIT DOA estimation method is analogous to the ESPRIT frequency estimator.

The ESPRIT DOA estimation method, like the ESPRIT frequency estimator, computes the DOA estimates by solving an $n \times n$ eigenvalue problem. *There is no search involved*, in contrast to the previous methods; in addition, *there is no problem of separating the "signal DOAs" from the "noise DOAs,"* once again in contrast to the Yule–Walker, MUSIC, and Min–Norm methods. However, unlike these other methods, *ESPRIT can only be used with the special array configuration described earlier*. In particular, this requirement limits the number of resolvable sources at $n < \bar{m}$ (as both A_1 and A_2 must have full column rank). Note that *the two subarrays do not need to be calibrated*, although they need to be identical, and *ESPRIT can be sensitive to differences between the two subarrays* in the same way that Yule–Walker, MUSIC, and Min–Norm are sensitive to imperfections in array calibration. Finally, note that, like the other DOA-finding algorithms presented in this section (with the exception of the NLS method), ESPRIT is not usable in the case of coherent signals.

6.5 COMPLEMENTS

6.5.1 On the Minimum–Norm Constraint

As explained in Section 6.4.4, the Root Min–Norm (temporal) frequency estimator, introduced in Section 4.6, can be used without modification for DOA estimation with a uniform linear array. Using the definitions and notation in Section 4.6, let $\hat{g} = [1\ \hat{g}_1 \ldots \hat{g}_{m-1}]^T$ denote the vector in

$\mathcal{R}(\hat{G})$ that has the first element equal to one and minimum Euclidean norm. Then, the Root Min–Norm DOA estimates are obtained from the roots of the polynomial

$$\hat{g}(z) = 1 + \hat{g}_1 z^{-1} + \cdots + \hat{g}_{m-1} z^{-(m-1)} \tag{6.5.1}$$

that are located nearest the unit circle. (See the description of Min–Norm in Section 4.6.) As N increases, the polynomial in (6.5.1) approaches

$$g(z) = 1 + g_1 z^{-1} + \cdots + g_{m-1} z^{-(m-1)} \tag{6.5.2}$$

where $g = [1 \; g_1 \ldots g_{m-1}]^T$ is the minimum–norm vector in $\mathcal{R}(G)$. In this complement, we show that (6.5.2) has n zeroes at $\{e^{-i\omega_k}\}_{k=1}^n$ (the so-called "signal zeroes") and $(m - n - 1)$ extraneous zeroes situated *strictly inside* the unit circle (the latter are normally called "noise zeroes"); here $\{\omega_k\}_{k=1}^n$ either are temporal frequencies or are spatial frequencies (as in (6.2.27)).

Let $g = [1, g_1, \ldots, g_{m-1}]^T \in \mathcal{R}(G)$. Then (4.2.4) and (4.5.6) imply that

$$a^*(\omega_k) \begin{bmatrix} 1 \\ g_1 \\ \vdots \\ g_{m-1} \end{bmatrix} = 0 \qquad \Longleftrightarrow$$

$$1 + g_1 e^{i\omega_k} + \cdots + g_{m-1} e^{i(m-1)\omega_k} = 0 \qquad (\text{for } k = 1, \ldots, n) \tag{6.5.3}$$

Hence, any polynomial $g(z)$ whose coefficient vector belongs to $\mathcal{R}(G)$ must have zeroes at $\{e^{-i\omega_k}\}_{k=1}^n$, and thus it can be factored as

$$g(z) = g_s(z) g_n(z) \tag{6.5.4}$$

where

$$g_s(z) = \prod_{k=1}^{n} (1 - e^{-i\omega_k} z^{-1})$$

The $(m - n - 1)$-degree polynomial $g_n(z)$ in (6.5.4) contains the noise zeroes, and at this point is arbitrary. (As the coefficients of $g_n(z)$ vary, the vectors made from the corresponding coefficients of $g(z)$ span $\mathcal{R}(G)$.)

Next, assume that g satisfies the minimum–norm constraint:

$$\sum_{k=0}^{m-1} |g_k|^2 = \min \qquad (g_0 \triangleq 1) \tag{6.5.5}$$

By using Parseval's theorem (see (1.2.6)), we can rewrite (6.5.5) as

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |g(\omega)|^2 \, d\omega = \min \Longleftrightarrow \frac{1}{2\pi} \int_{-\pi}^{\pi} |g_n(\omega)|^2 \; |g_s(\omega)|^2 \, d\omega = \min \tag{6.5.6}$$

(where, by convention, $g(\omega) = g(z)\big|_{z=e^{i\omega}}$). Since $g_s(z)$ in (6.5.4) is fixed, the minimization in (6.5.6) is over $g_n(z)$.

To proceed, some additional notation is required. Let

$$g_n(z) = 1 + \alpha_1 z^{-1} + \cdots + \alpha_{m-n-1} z^{-(m-n-1)}$$

and let $y(t)$ be a signal whose PSD is equal to $|g_s(\omega)|^2$; hence, $y(t)$ is an nth-order MA process. By making use of (1.3.9) and (1.4.9), along with the previous notation, we can write (6.5.6) in the following equivalent form:

$$\min_{\{\alpha_k\}} E\left\{|y(t) + \alpha_1 y(t-1) + \cdots + \alpha_{m-n-1} y(t-m+n+1)|^2\right\} \tag{6.5.7}$$

The minimizing coefficients $\{\alpha_k\}$ are given by the solution to a Yule–Walker system of equations similar to (3.4.6). (To show this, parallel the calculation leading to (3.4.8) and (3.4.12).) Since the covariance matrix, of any finite dimension, associated with a moving-average signal is positive definite, it follows that

- the coefficients $\{\alpha_k\}$, and hence $\{g_k\}$, are *uniquely* determined by the minimum–norm constraint
- the polynomial $g_n(z)$ whose coefficients are obtained from (6.5.7) has all its zeroes *strictly inside* the unit circle (*cf.* Exercise 3.8)

which was to be proven.

Thus, the choice of $\hat{g}$ in the Min–Norm algorithm makes it possible to separate the signal zeroes from the noise zeroes, at least for data samples that are sufficiently long. (For small or medium-sized samples, it might happen that noise zeroes get closer to the unit circle than signal zeroes, which would lead to spurious frequency or DOA estimates.)

As a final remark, note, from (6.5.6), that there is little reason for $g_n(z)$ to have zeroes in the sectors where the signal zeroes are present (since the integrand in (6.5.6) is already quite small for ω values close to $\{\omega_k\}_{k=1}^n$). Hence, we can expect the extraneous zeroes to be more or less uniformly distributed inside the unit circle, in sectors that do not contain signal zeroes (see, e.g., [Kumaresan 1983].)

For more details on the topic of this complement, see [Tufts and Kumaresan 1982; Kumaresan 1983].

6.5.2 NLS Direction-of-Arrival Estimation for a Constant-Modulus Signal

The NLS estimation of the DOA of a *single signal* impinging on an array of sensors is obtained by minimizing the criterion (6.4.4) with $n = 1$,

$$\sum_{t=1}^{N} \|y(t) - a(\theta)s(t)\|^2 \tag{6.5.8}$$

with respect to $\{s(t)\}_{t=1}^{N}$ and θ. The result is obtained from equation (6.4.7), which for $n = 1$ reduces to

$$\hat{\theta} = \arg\max_{\theta} a^*(\theta)\hat{R}a(\theta) = \arg\max_{\theta} \sum_{t=1}^{N} \left|a^*(\theta)y(t)\right|^2 \tag{6.5.9}$$

This, of course, is nothing but the *beamforming DOA estimate* for $n = 1$; see (6.3.18). Hence, as expected (see the Remark following (6.4.7) and also (4.3.11)), the NLS estimate of the DOA of an *arbitrary* signal coincides with the beamforming estimate.

In this complement, we will solve the NLS direction-of-arrival estimation problem in (6.5.8), under the assumption that $\{s(t)\}$ is a *constant-modulus signal*; that is,

$$s(t) = \alpha e^{i\phi(t)} \tag{6.5.10}$$

where $\alpha > 0$ denotes the unknown signal amplitude and $\{\phi(t)\}$ is its unknown phase sequence. We assume $\alpha > 0$ to avoid a phase ambiguity in $\{\phi(t)\}$. Signals of this type are often encountered in communication applications with phase-modulated waveforms.

Inserting (6.5.10) in (6.5.8) yields the following criterion that is to be minimized with respect to $\{\phi(t)\}_{t=1}^{N}$, α, and θ:

$$\begin{aligned} &\sum_{t=1}^{N} \left\|y(t) - \alpha e^{i\phi(t)} a(\theta)\right\|^2 \\ &\quad = \sum_{t=1}^{N} \left\{\|y(t)\|^2 + \alpha^2\|a(\theta)\|^2 - 2\alpha \operatorname{Re}\left[a^*(\theta)y(t)e^{-i\phi(t)}\right]\right\} \end{aligned} \tag{6.5.11}$$

It follows from (6.5.11) that the NLS estimate of $\{\phi(t)\}_{t=1}^{N}$ is given by the maximizer of the function

$$\begin{aligned} \operatorname{Re}\left[a^*(\theta)y(t)e^{-i\phi(t)}\right] &= \operatorname{Re}\left[\left|a^*(\theta)y(t)\right| e^{i\arg[a^*(\theta)y(t)]}e^{-i\phi(t)}\right] \\ &= \left|a^*(\theta)y(t)\right| \cos\left[\arg\big(a^*(\theta)y(t)\big) - \phi(t)\right] \end{aligned} \tag{6.5.12}$$

which is easily seen to be

$$\boxed{\hat{\phi}(t) = \arg\left[a^*(\theta)y(t)\right], \qquad t = 1, \ldots, N} \tag{6.5.13}$$

From (6.5.11)–(6.5.13), along with the assumption that $\|a(\theta)\|$ is constant (which is also used to derive (6.5.9)), we can readily verify that the NLS estimate of θ for the constant modulus signal case is given by

$$\boxed{\hat{\theta} = \arg\max_{\theta} \sum_{t=1}^{N} \left|a^*(\theta)y(t)\right|} \tag{6.5.14}$$

Finally, the NLS estimate of α is obtained by minimizing (6.5.11) (with $\{\phi(t)\}$ and θ replaced by (6.5.13) and (6.5.14), respectively):

$$\hat{\alpha} = \frac{1}{N\|a(\hat{\theta})\|^2} \sum_{t=1}^{N} \left| a^*(\hat{\theta})y(t) \right| \tag{6.5.15}$$

Remark: It follows easily from the preceding derivation that, if α is known (as could be the case when the emitted signal has a known amplitude that is not significantly distorted during propagation), the NLS estimates of θ and $\{\phi(t)\}$ are still given by (6.5.13) and (6.5.14). ■

Interestingly, the only difference between the beamformer for an arbitrary signal, (6.5.9), and the beamformer for a constant-modulus signal, (6.5.14), is that *the "squaring operation" is missing in the latter*. This difference is somewhat analogous to the one pointed out in Complement 4.9.4, even though the models considered there and in this complement are rather different from one another.

For more details on the subject of this complement, see [Stoica and Besson 2000] and its references.

6.5.3 Capon Method: Further Insights and Derivations

The spatial filter (or beamformer) used in the beamforming method is independent of data. In contrast, the Capon spatial filter is data dependent, or *data adaptive*; see equation (6.3.24). It is this data adaptivity that confers on the Capon method better resolution and significantly reduced leakage as compared with the beamforming method.

An interesting fact about the Capon method for temporal or spatial spectral analysis is that it can be derived in several ways. The standard derivation is given in Section 6.3.2. This complement presents four additional derivations of the Capon method, each not as well known as the standard derivation. Each of the derivations presented here is based on an intuitively appealing design criterion. Collectively, they provide further insights into the features and possible interpretations of the Capon method.

APES-Like Derivation

Let θ denote a generic DOA, and consider equation (6.2.19),

$$y(t) = a(\theta)s(t) + e(t) \tag{6.5.16}$$

which describes the array output, $y(t)$, as the sum of a possible signal component impinging from the generic DOA θ and a term $e(t)$ that includes noise and any other signals with DOAs different from θ. Let σ_s^2 denote the power of the signal $s(t)$ in (6.5.16), which is the main parameter we want to estimate: σ_s^2 as a function of θ provides an estimate of the spatial spectrum. Let us estimate the spatial filter vector, h, as well as the signal power, σ_s^2, by solving the following least-squares (LS) problem:

$$\boxed{\min_{h,\sigma_s^2} E\left\{|h^*y(t) - s(t)|^2\right\}} \tag{6.5.17}$$

Of course, the signal $s(t)$ in (6.5.17) is not known. However, as we show shortly, (6.5.17) does not depend on $s(t)$ but only on its power σ_s^2, so the fact that $s(t)$ in (6.5.17) is unknown does not pose a problem. Also, note that the vector h in (6.5.17) is *not* constrained, as it is in (6.3.24).

Assuming that $s(t)$ in (6.5.16) is uncorrelated with the noise-plus-interference term $e(t)$, we obtain

$$E\left\{y(t)s^*(t)\right\} = a(\theta)\sigma_s^2 \tag{6.5.18}$$

which implies that

$$\begin{aligned} E\left\{|h^*y(t) - s(t)|^2\right\} &= h^*Rh - h^*a(\theta)\sigma_s^2 - a^*(\theta)h\sigma_s^2 + \sigma_s^2 \\ &= \left[h - \sigma_s^2 R^{-1}a(\theta)\right]^* R\left[h - \sigma_s^2 R^{-1}a(\theta)\right] \\ &\quad + \sigma_s^2\left[1 - \sigma_s^2 a^*(\theta)R^{-1}a(\theta)\right] \end{aligned} \tag{6.5.19}$$

Omitting the trivial solution ($h = 0, \sigma_s^2 = 0$), the minimization of (6.5.19) with respect to h and σ_s^2 yields

$$h = \frac{R^{-1}a(\theta)}{a^*(\theta)R^{-1}a(\theta)} \tag{6.5.20}$$

$$\sigma_s^2 = \frac{1}{a^*(\theta)R^{-1}a(\theta)} \tag{6.5.21}$$

which coincides with the Capon solution in (6.3.24) and (6.3.25). To obtain σ_s^2 in (6.5.21), we used the fact that the criterion in (6.5.19) should be greater than or equal to zero for any h and σ_s^2.

The LS-fitting criterion in (6.5.17) is *reminiscent of the APES approach* discussed in Complement 5.6.4. The use of APES for array processing is discussed in Complement 6.5.6, under the assumption that $\{s(t)\}$ is an unknown *deterministic* sequence. Interestingly, using the APES design principle in the above manner, under the assumption that the signal $s(t)$ in (6.5.16) is *stochastic*, leads to the Capon method.

Inverse-Covariance-Fitting Derivation

The covariance matrix of the signal term $a(\theta)s(t)$ in (6.5.16) is given by

$$\sigma_s^2 a(\theta)a^*(\theta) \tag{6.5.22}$$

We can obtain the beamforming method (see Section 6.3.1) by fitting (6.5.22) to R in a least-squares sense:

$$\begin{aligned} &\min_{\sigma_s^2} \|R - \sigma_s^2 a(\theta)a^*(\theta)\|^2 \\ &\quad = \min_{\sigma_s^2}\{\text{constant} + \sigma_s^4[a^*(\theta)a(\theta)]^2 - 2\sigma_s^2 a^*(\theta)Ra(\theta)\} \end{aligned} \tag{6.5.23}$$

Because $a^*(\theta)a(\theta) = m$ (by assumption; see (6.3.11)), it follows from (6.5.23) that the minimizing σ_s^2 is given by

$$\sigma_s^2 = \frac{1}{m^2} a^*(\theta) R a(\theta) \tag{6.5.24}$$

which coincides with the beamforming estimate of the power coming from DOA θ (see (6.3.16)).

To obtain the Capon method by following an idea similar to the one above, we fit the pseudoinverse of (6.5.22) to the inverse of R:

$$\boxed{\min_{\sigma_s^2} \left\| R^{-1} - \left[\sigma_s^2 a(\theta) a^*(\theta)\right]^\dagger \right\|^2} \tag{6.5.25}$$

It is easily verified that the Moore–Penrose pseudoinverse of $\sigma_s^2 a(\theta)a^*(\theta)$ is given by

$$\left[\sigma_s^2 a(\theta)a^*(\theta)\right]^\dagger = \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{[a^*(\theta)a(\theta)]^2} = \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{m^2} \tag{6.5.26}$$

This follows, for instance, from (A.8.8) and the fact that

$$\sigma_s^2 a(\theta)a^*(\theta) = \left[\sigma_s^2 \|a(\theta)\|^2\right] \left[\frac{a(\theta)}{\|a(\theta)\|}\right] \left[\frac{a(\theta)}{\|a(\theta)\|}\right]^* \triangleq \sigma u v^* \tag{6.5.27}$$

is the singular-value decomposition (SVD) of $\sigma_s^2 a(\theta)a^*(\theta)$. Inserting (6.5.26) into (6.5.25) leads to the problem

$$\min_{\sigma_s^2} \left\| R^{-1} - \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{m^2} \right\|^2 \tag{6.5.28}$$

whose solution, by analogy with (6.5.23)–(6.5.24), is given by the Capon estimate of the signal power:

$$\sigma_s^2 = \frac{1}{a^*(\theta)R^{-1}a(\theta)} \tag{6.5.29}$$

It is worth noting that in the present *covariance-fitting-based derivation*, the signal power σ_s^2 is estimated directly *without* the need to first obtain an intermediate spatial filter h. The remaining two derivations of the Capon method are of the same type.

Weighted-Covariance-Fitting Derivation

The least-squares criterion in (6.5.23), which yields the beamforming method, does not take into account the fact that the sample estimates of the different elements of the data covariance matrix do not have the same accuracy. It was shown (e.g., in [Ottersten, Stoica, and Roy 1998] and its references) that the following *weighted LS covariance-fitting criterion* takes the accuracies of the different elements of the sample covariance matrix into account in an *optimal manner*:

$$\boxed{\min_{\sigma_s^2} \left\| R^{-1/2} \left[R - \sigma_s^2 a(\theta) a^*(\theta) \right] R^{-1/2} \right\|^2} \tag{6.5.30}$$

Here, $R^{-1/2}$ denotes the Hermitian square root of R^{-1}. By a straightforward calculation, we can rewrite the criterion in (6.5.30) in the following equivalent form:

$$\begin{aligned} &\left\| I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2} \right\|^2 \\ &\quad = \text{constant} - 2\sigma_s^2 a^*(\theta) R^{-1} a(\theta) + \sigma_s^4 \left[a^*(\theta) R^{-1} a(\theta) \right]^2 \end{aligned} \tag{6.5.31}$$

The minimization of (6.5.31) with respect to σ_s^2 yields

$$\sigma_s^2 = \frac{1}{a^*(\theta) R^{-1} a(\theta)}$$

which coincides with the Capon solution in (6.3.26).

Constrained-Covariance-Fitting Derivation

The final derivation of the Capon method that we will present is also based on a covariance-fitting criterion, but in a manner quite different from those in the previous two derivations. Our goal here is still to obtain the signal power by fitting $\sigma_s^2 a(\theta) a^*(\theta)$ to R, but now we explicitly impose the condition that the residual covariance matrix, $R - \sigma_s^2 a(\theta) a^*(\theta)$, should be positive semidefinite, and we "minimize" the approximation (or fitting) error by choosing the maximum possible value of σ_s^2 for which this condition holds. Mathematically, σ_s^2 is the solution to the following constrained covariance-fitting problem:

$$\boxed{\max_{\sigma_s^2} \sigma_s^2 \quad \text{subject to } R - \sigma_s^2 a(\theta) a^*(\theta) \geq 0} \tag{6.5.32}$$

The solution to (6.5.32) can be obtained in the following way, which is a simplified version of the original derivation in [Marzetta 1983]: Let $R^{-1/2}$ again denote the Hermitian square root of R^{-1}. Then the following equivalences can be readily verified:

$$\begin{aligned} &R - \sigma_s^2 a(\theta) a^*(\theta) \geq 0 \\ &\quad \Longleftrightarrow I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2} \geq 0 \\ &\quad \Longleftrightarrow 1 - \sigma_s^2 a^*(\theta) R^{-1} a(\theta) \geq 0 \\ &\quad \Longleftrightarrow \sigma_s^2 \leq \frac{1}{a^*(\theta) R^{-1} a(\theta)} \end{aligned} \tag{6.5.33}$$

The third line in equation (6.5.33) follows from the fact that the eigenvalues of the matrix $I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2}$ are equal to 1 minus the eigenvalues of $\sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2}$ (see Result R5 in Appendix A), and the latter eigenvalues are given by $\sigma_s^2 a^*(\theta) R^{-1} a(\theta)$ (which is the trace of the previous matrix) along with $(m-1)$ zeroes. From (6.5.33), we can see that the Capon spectral estimate is the solution to the problem (6.5.32) as well.

The equivalence between the formulation of the Capon method in (6.5.32) and the standard formulation in Section 6.3.2 can also be shown as follows: The constraint in (6.5.32) is equivalent to the requirement that

$$h^*[R - \sigma_s^2 a(\theta) a^*(\theta)]\ h \geq 0 \text{ for any } h \in \mathbf{C}^{m \times 1} \tag{6.5.34}$$

which, in turn, is equivalent to

$$\begin{gathered} h^*[R - \sigma_s^2 a(\theta) a^*(\theta)]\ h \geq 0 \\ \text{for any } h \text{ such that } h^* a(\theta) = 1 \end{gathered} \tag{6.5.35}$$

Clearly, (6.5.34) implies (6.5.35). To also show that (6.5.35) implies (6.5.34), let h be such that $h^* a(\theta) = \alpha \neq 0$; then h/α^* satisfies $(h/\alpha^*)^* a(\theta) = 1$ and, hence, by the assumption that (6.5.35) holds,

$$\frac{1}{|\alpha|^2} h^*[R - \sigma_s^2 a(\theta) a^*(\theta)] h \geq 0$$

which shows that (6.5.35) implies (6.5.34) for any h satisfying $h^* a(\theta) \neq 0$. Now, if h is such that $h^* a(\theta) = 0$, then

$$h^*[R - \sigma_s^2 a(\theta) a^*(\theta)] h = h^* R h \geq 0$$

because $R > 0$ by assumption. This observation concludes the proof that (6.5.34) is equivalent to (6.5.35).

Using the equivalence of (6.5.34) and (6.5.35), we can rewrite (6.5.34) as follows:

$$h^* R h \geq \sigma_s^2 \quad \text{for any } h \text{ such that } h^* a(\theta) = 1 \tag{6.5.36}$$

From (6.5.36), we can see that the solution to (6.5.32) is given by

$$\sigma_s^2 = \min_h h^* R h \quad \text{subject to } h^* a(\theta) = 1$$

which coincides with the standard formulation of the Capon method in (6.3.24).

The formulation of the Capon method in (6.5.32) will be used in Complement 6.5.4 to extend the method to the case where the direction vector $a(\theta)$ is known only imprecisely.

6.5.4 Capon Method for Uncertain Direction Vectors

The Capon method has better resolution and much better interference rejection capability (i.e., much lower leakage) than the beamforming method, *provided that the direction vector, $a(\theta)$, is* *known accurately.* However, whenever the knowledge of $a(\theta)$ is imprecise, the performance of the Capon method could become worse than that of the beamforming method. To see why this is so, consider a scenario in which the problem is to estimate the power coming from a source with DOA assumed to be equal to θ_0. Let us assume that, in actuality, the true DOA of the source is $\theta_0 + \Delta$. For the Capon beamformer pointed toward θ_0, the source of interest (located at $\theta_0 + \Delta$) will play the role of an interference and will be attenuated. Consequently, the power of the signal of interest will be underestimated; the larger Δ is, the larger the underestimation error. Because steering-vector errors are common in applications, it follows that a *robust version of the Capon method* (i.e., one that is as insensitive to steering-vector errors as possible) would be highly desirable.

In this complement, we will present *an extension of the Capon method to the case of uncertain direction vectors*. Specifically, we will assume that the only knowledge we have about $a(\theta)$ is that it belongs to the uncertainty ellipsoid

$$(a - \bar{a})^* C^{-1} (a - \bar{a}) \leq 1 \tag{6.5.37}$$

where the vector $\bar{a}$ and the positive definite matrix C are given. Note that both a and $\bar{a}$, as well as C, usually depend on θ; however, for the sake of notational convenience, we drop the θ dependence of these variables.

In some applications, there will be too little available information about the errors in the steering vector to make a competent choice of the full matrix C in (6.5.37). In such cases, we may simply set $C = \varepsilon I$, so that (6.5.37) becomes

$$\|a - \bar{a}\|^2 \leq \varepsilon \tag{6.5.38}$$

where ε is a positive number. Let a_0 denote the true (and unknown) direction vector, and let $\varepsilon_0 = \|a_0 - \bar{a}\|^2$ where, as before, $\bar{a}$ is the assumed direction vector. Ideally, we should choose $\varepsilon = \varepsilon_0$. However, it can be shown (see [Stoica, Wang, and Li 2003], [Li, Stoica, and Wang 2003]) that the performance of the robust Capon method remains almost unchanged when ε is varied in a relatively large interval around ε_0.

As already stated, our goal here is to obtain a robust Capon method that is insensitive to errors in the direction (or steering) vector. We will do so by combining the covariance-fitting formulation in (6.5.32) for the standard Capon method with the steering uncertainty set in (6.5.37). Hence, we aim to derive estimates of *both* σ_s^2 *and* a by solving the following constrained covariance-fitting problem:

$$\boxed{\begin{array}{lll} \max\limits_{a,\sigma_s^2} \sigma_s^2 & \text{subject to:} & R - \sigma_s^2 a a^* \geq 0 \\ & & (a - \bar{a})^* C^{-1} (a - \bar{a}) \leq 1 \end{array}} \tag{6.5.39}$$

To avoid the trivial solution ($a \to 0, \sigma_s^2 \to \infty$), we assume that $a = 0$ does not belong to the uncertainty ellipsoid in (6.5.39), or, equivalently, that

$$\bar{a}^* C^{-1} \bar{a} > 1 \tag{6.5.40}$$

(which is a regularity condition).

Because both σ_s^2 and a are considered to be free parameters in the previous fitting problem, there is a scaling ambiguity in the signal covariance term in (6.5.39), in the sense that both (σ_s^2, a) and $(\sigma_s^2/\mu, \mu^{1/2}a)$ for any $\mu > 0$ give the same covariance term $\sigma_s^2 aa^*$. To eliminate this ambiguity, we can use the knowledge that the true steering vector satisfies the following condition (see (6.3.11)):

$$a^*a = m \tag{6.5.41}$$

However, the constraint in (6.5.41) is nonconvex and so makes the combined problem (6.5.39) and (6.5.41) somewhat more difficult to solve than (6.5.39). On the other hand, (6.5.39) (without (6.5.41)) can be solved quite efficiently, as we show next. To take advantage of this fact, we can make use of (6.5.41) to eliminate the scaling ambiguity in the following pragmatic way:

- Obtain the solution $(\tilde{\sigma}_s^2, \tilde{a})$ of (6.5.39).
- Obtain an estimate of a that satisfies (6.5.41) by scaling $\tilde{a}$,

  $$\hat{a} = \frac{\sqrt{m}}{\|\tilde{a}\|}\tilde{a}$$

  and a corresponding estimate of σ_s^2 by scaling $\tilde{\sigma}_s^2$, such that the signal covariance term is left unchanged (i.e., $\tilde{\sigma}_s^2\tilde{a}\tilde{a}^* = \hat{\sigma}_s^2\hat{a}\hat{a}^*$), which gives

  $$\hat{\sigma}_s^2 = \tilde{\sigma}_s^2\frac{\|\tilde{a}\|^2}{m} \tag{6.5.42}$$

To derive the solution $(\tilde{\sigma}_s^2, \tilde{a})$ of (6.5.39), we first note that, for any fixed a, the maximizing σ_s^2 is given by

$$\tilde{\sigma}_s^2 = \frac{1}{a^*R^{-1}a} \tag{6.5.43}$$

(See equation (6.5.33) in Complement 6.5.3.) This simple observation allows us to eliminate σ_s^2 from (6.5.39) and hence reduce (6.5.39) to the following problem:

$$\min_a a^*R^{-1}a \quad \text{subject to: } (a-\bar{a})^*C^{-1}(a-\bar{a}) \le 1 \tag{6.5.44}$$

Under the regularity condition in (6.5.40), the solution $\tilde{a}$ to (6.5.44) will occur on the boundary of the constraint set; therefore, we can reformulate (6.5.44) as the following quadratic problem with a quadratic equality constraint:

$$\min_a a^*R^{-1}a \quad \text{subject to: } (a-\bar{a})^*C^{-1}(a-\bar{a}) = 1 \tag{6.5.45}$$

This problem can be solved efficiently by using the Lagrange multiplier approach—see [Li, Stoica, and Wang 2003]. In the remaining part of this complement, we derive the Lagrange-multiplier solver in [Li, Stoica, and Wang 2003], but in a more self-contained way.

To simplify the notation, consider (6.5.45) with $C = \varepsilon I$ as in (6.5.38):

$$\min_a a^* R^{-1} a \quad \text{subject to: } \|a - \bar{a}\|^2 = \varepsilon \tag{6.5.46}$$

(The case of $C \neq \varepsilon I$ can be treated similarly.) Define

$$x = a - \bar{a} \tag{6.5.47}$$

and rewrite (6.5.46), using x in lieu of a:

$$\min_x \left[x^* R^{-1} x + x^* R^{-1} \bar{a} + \bar{a}^* R^{-1} x \right] \quad \text{subject to: } \|x\|^2 = \varepsilon \tag{6.5.48}$$

The constraint in (6.5.48) makes the x that solves (6.5.48) also a solution to the problem

$$\min_x \left[x^* (R^{-1} + \lambda I) x + x^* R^{-1} \bar{a} + \bar{a}^* R^{-1} x \right] \quad \text{subject to: } \|x\|^2 = \varepsilon \tag{6.5.49}$$

where λ is an arbitrary constant. Let us consider a particular choice of λ that is a solution of the equation

$$\bar{a}^* (I + \lambda R)^{-2} \bar{a} = \varepsilon \tag{6.5.50}$$

and is also such that

$$R^{-1} + \lambda I > 0 \tag{6.5.51}$$

Then, the *unconstrained* minimizer of the function in (6.5.49) is given by

$$x = -\left(R^{-1} + \lambda I\right)^{-1} R^{-1} \bar{a} = -(I + \lambda R)^{-1} \bar{a} \tag{6.5.52}$$

and it satisfies the constraint in (6.5.49) (*cf.* (6.5.50)). It follows that x in (6.5.52) with λ given by (6.5.50) and (6.5.51) is the solution to (6.5.49) (and, hence, to (6.5.48)). Hence, what is left to explain is how to solve (6.5.50) under the condition (6.5.51) in an efficient manner; we will do that next.

Let

$$R = U \Lambda U^* \tag{6.5.53}$$

denote the eigenvalue decomposition (EVD) of R, where $U^* U = U U^* = I$ and

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}; \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \tag{6.5.54}$$

Also, let

$$b = U^* \bar{a} \tag{6.5.55}$$

Using (6.5.53)–(6.5.55), we can rewrite the left-hand side of equation (6.5.50) as

$$\begin{aligned} g(\lambda) &\triangleq \bar{a}^* [I + \lambda R]^{-2} \bar{a} = \bar{a}^* \left[U(I + \lambda\Lambda)U^*\right]^{-2} \bar{a} \\ &= b^*(I + \lambda\Lambda)^{-2} b = \sum_{k=1}^{m} \frac{|b_k|^2}{(1 + \lambda\lambda_k)^2} \end{aligned} \tag{6.5.56}$$

where b_k is the kth element of the vector b. Note that

$$\sum_{k=1}^{m} |b_k|^2 = \|b\|^2 = \|\bar{a}\|^2 > \varepsilon \tag{6.5.57}$$

(See (6.5.55) and (6.5.40).) It follows from (6.5.56) and (6.5.57) that λ can be a solution of the equation $g(\lambda) = \varepsilon$ only if

$$(1 + \lambda\lambda_k)^2 > 1 \tag{6.5.58}$$

for some value of k. At the same time, λ should (see (6.5.51)) be such that

$$\begin{aligned} R^{-1} + \lambda I > 0 &\iff I + \lambda R > 0 \\ &\iff 1 + \lambda\lambda_k > 0 \text{ for } k = 1, \ldots, m \end{aligned} \tag{6.5.59}$$

It follows from (6.5.58) and (6.5.59) that $1 + \lambda\lambda_k > 1$ for at least one value of k, which implies that

$$\lambda > 0 \tag{6.5.60}$$

This inequality sets a lower bound on the solution to (6.5.50). To refine this lower bound, and also to obtain an upper bound, first observe that $g(\lambda)$ is a *monotonically decreasing function of* λ for $\lambda > 0$. Furthermore, for

$$\lambda_L = \frac{\|\bar{a}\| - \sqrt{\varepsilon}}{\lambda_1 \sqrt{\varepsilon}} \tag{6.5.61}$$

we have that

$$g(\lambda_L) \geq \frac{1}{(1 + \lambda_L\lambda_1)^2} \|b\|^2 = \frac{\varepsilon}{\|\bar{a}\|^2} \|\bar{a}\|^2 = \varepsilon \tag{6.5.62}$$

Similarly, for

$$\lambda_U = \frac{\|\bar{a}\| - \sqrt{\varepsilon}}{\lambda_m \sqrt{\varepsilon}} \geq \lambda_L \tag{6.5.63}$$

we can verify that

$$g(\lambda_U) \leq \frac{1}{(1 + \lambda_U\lambda_m)^2} \|b\|^2 = \varepsilon \tag{6.5.64}$$

Summarizing the previous facts, it follows that *equation* (6.5.50) *has a unique solution for* λ *that satisfies (6.5.51), which belongs to the interval* $[\lambda_L, \lambda_U] \subset (0, \infty)$. With this observation, the derivation of the robust version of the Capon method is complete. The following is a *step-by-step summary of the Robust Capon algorithm.*

The Robust Capon Algorithm

Step 1. Compute the eigendecomposition $R = U \Lambda U^*$, and set $b = U^* \bar{a}$.

Step 2. Solve the equation $g(\lambda) = \varepsilon$ for λ (using, e.g., a Newton method, along with the fact that there is a unique solution in the interval $[\lambda_L, \lambda_U]$).

Step 3. Compute (*cf.* (6.5.47), (6.5.52), (6.5.53))

$$\tilde{a} = \bar{a} - U(I + \lambda \Lambda)^{-1} b \tag{6.5.65}$$

and, finally, compute the power estimate (see (6.5.42) and (6.5.43))

$$\hat{\sigma}_s^2 = \frac{\tilde{a}^* \tilde{a}}{m\, \tilde{a}^* U \Lambda^{-1} U^* \tilde{a}} \tag{6.5.66}$$

where, from (6.5.65), $U^* \tilde{a} = b - (I + \lambda \Lambda)^{-1} b$.

The bulk of the computation in the algorithm involves computing the EVD of R, which requires $\mathcal{O}(m^3)$ arithmetic operations. Hence, the computational complexity of the Robust Capon method is comparable to that of the standard Capon method. We refer the reader to [Li, Stoica, and Wang 2003] and also to [Stoica, Wang, and Li 2003] for further computational considerations and insights, and for many numerical examples illustrating the good performance of the Robust Capon method, including its insensitivity to the choice of ε in (6.5.38) or C in (6.5.37).

6.5.5 Capon Method with Noise-Gain Constraint

As was explained in Complement 6.5.4, the Capon method performs poorly as a power estimator in the presence of steering-vector errors. (Yet, it could perform fairly well as a DOA estimator, provided that the SNR is reasonably large; see [Cox 1973; Li, Stoica, and Wang 2003] and references therein.) The same happens when the number of snapshots, N, is relatively small, such as when N is equal to or only slightly larger than the number of sensors, m. In fact, there is a close relationship between the cases of steering-vector errors and those of small-sample errors—see, for example, [Feldman and Griffiths 1994]. More precisely, the sampling estimation errors of the covariance matrix can be viewed as steering-vector errors in a corresponding theoretical covariance matrix, and vice versa. For example, consider a uniform linear array, and assume that the source signals are uncorrelated with one another. In this case, the theoretical covariance matrix R of the array output is Toeplitz. Assume that the sample covariance matrix $\hat{R}$ is also Toeplitz. According to the Carathéodory parameterization of Toeplitz matrices (see Complement 4.9.2), we can view $\hat{R}$ as being the *theoretical* covariance matrix associated with a fictitious ULA on which uncorrelated signals impinge, but the powers and DOAs of the latter signals are different from those of the actual signals. Hence, the small sample estimation errors in $\hat{R}$ can be viewed as being due to steering-vector errors in a corresponding theoretical covariance matrix.

The robust Capon method (RCM) presented in Complement 6.5.4 significantly outperforms the standard Capon method (CM) in power-estimation applications in which the sample length is insufficient for accurate estimation of R or in which the steering vector is known imprecisely. The RCM was introduced in [Stoica, Wang, and Li 2003; Li, Stoica, and Wang 2003]. An earlier approach, whose goal is also to enhance the performance of CM in the presence of sampling-estimation errors or steering-vector mismatch, is the so-called *diagonal-loading* approach (see, e.g., [Hudson 1981; Van Trees 2002] and references therein). The main idea of diagonal loading is to replace R in the Capon formula for the spatial filter h, (6.3.24), by the matrix

$$R + \lambda I \tag{6.5.67}$$

where the diagonal-loading factor $\lambda > 0$ is a user-selected parameter. The filter vector h so obtained is given by

$$h = \frac{(R + \lambda I)^{-1} a}{a^*(R + \lambda I)^{-1} a} \tag{6.5.68}$$

The use of the diagonally loaded matrix in (6.5.67) instead of R is the reason for the name of the approach based on (6.5.68). The symbol R in this complement refers either to a theoretical covariance matrix or to a sample covariance matrix.

There have been several rules proposed in the literature for choosing the parameter λ in (6.5.68). Most of these rules choose λ in a rather *ad hoc* and data-independent manner. As illustrated in [Li, Stoica, and Wang 2003] and its references, a data-independent selection of the diagonal-loading factor *cannot* improve the performance for a reasonably large range of SNR values. Hence, a *data-dependent choice of* λ is desired.

One commonly used data-dependent rule selects the diagonal-loading factor $\lambda > 0$ that satisfies

$$\|h\|^2 = \frac{a^*(R + \lambda I)^{-2} a}{\left[a^*(R + \lambda I)^{-1} a\right]^2} = c \tag{6.5.69}$$

where the constant c must be chosen by the user. Let us explain briefly why choosing λ via (6.5.69) makes sense intuitively. Assume that the array output vector contains a spatially white noise component whose covariance matrix is proportional to I (see (6.4.1)). Then the power at the output of the spatial filter h due to the noise component is $\|h\|^2$; for this reason, $\|h\|^2$ is sometimes called *the (white) noise gain* of h. In scenarios with a large number of (possibly closely spaced) source signals, the Capon spatial filter h in (6.3.24) could run out of "degrees of freedom" and hence not pay enough attention to the noise in the data (unless the SNR is very low). The result is a relatively high noise gain, $\|h\|^2$, which may well degrade the accuracy of signal power estimation. To prevent this from happening, it makes sense to limit $\|h\|^2$ as in (6.5.69). By doing so, we are left with the problem of choosing c. The choice of c might be easier than the direct choice of λ in (6.5.68), yet it is far from trivial, and, in fact, clear-cut rules for selecting c are hardly available. In particular, a "too small" value of c could limit the noise gain unnecessarily and result in decreased resolution and increased leakage.

In this complement, we will show that the spatial filter of the diagonally loaded Capon method in (6.5.68), (6.5.69) is the solution to the following design problem:

$$\boxed{\min_h h^*Rh \quad \text{subject to: } h^*a = 1 \text{ and } \|h\|^2 \leq c} \tag{6.5.70}$$

Because (6.5.70) is obtained by adding the noise-gain constraint $\|h\|^2 \leq c$ to the standard Capon problem in (6.3.23), we will call the method that follows from (6.5.70) the *constrained Capon method* (CCM). The fact that (6.5.68), (6.5.69) is the solution to (6.5.70) is well known from the previous literature (see, e.g., [Hudson 1981]). However, we present a rigorous and more thorough analysis of this solution. As a by-product, the analysis that follows also suggests some guidelines for choosing the user parameter c in (6.5.69). Note that, in general, a, c, and h in (6.5.70) *depend on the DOA* θ; to simplify notation, we will omit the functional dependence on θ here.

It is interesting to observe that *the RCM, described in Complement 6.5.4, can also be cast into a diagonal-loading framework*. To see this, first note from (6.5.47) and (6.5.52) that the steering-vector estimate used in the RCM is given by

$$\begin{aligned} a &= \bar{a} - (I + \lambda R)^{-1}\bar{a} = (I + \lambda R)^{-1}\left[(I + \lambda R) - I\right]\bar{a} \\ &= \left(\frac{1}{\lambda}R^{-1} + I\right)^{-1}\bar{a} \end{aligned} \tag{6.5.71}$$

The RCM estimates the signal power by

$$\frac{1}{a^*R^{-1}a} \tag{6.5.72}$$

with a as given in (6.5.71); hence, RCM does not directly use any spatial filter. However, the power estimate in (6.5.72) is equal to h^*Rh, where

$$h = \frac{R^{-1}a}{a^*R^{-1}a} \tag{6.5.73}$$

hence, (6.5.72) can be viewed as being obtained by the (implicit) use of the spatial filter in (6.5.71), (6.5.73). Inserting (6.5.71) into (6.5.73), we obtain

$$h = \frac{\left(R + \frac{1}{\lambda}I\right)^{-1} a}{a^*\left[\left(R + \frac{1}{\lambda}I\right)R^{-1}\left(R + \frac{1}{\lambda}I\right)\right]^{-1} a} \tag{6.5.74}$$

which, except for the scalar in the denominator, has the form in (6.5.68) of the spatial filter used by the diagonal-loading approach. Note that the diagonal-loading factor, $1/\lambda$, in (6.5.74) is data dependent. Furthermore, the selection of λ in the RCM (see Complement 6.5.4 for details on this aspect) relies entirely on information about the uncertainty set of the steering vector, as defined, for instance, by the sphere with radius $\varepsilon^{1/2}$ in (6.5.38). Such information is more readily available in applications than is information that would help the user select the noise-gain constraint c in the CCM. Indeed, in many applications, we should be able to make *a more competent guess about* ε *than about* c (for all DOAs of interest in the analysis). This appears to be a significant advantage of RCM over CCM, despite the fact that both methods can be interpreted as data-dependent diagonal-loading approaches.

Remark: The reader might have noted by now that the CCM problem in (6.5.70) is similar to the combined RCM problem in (6.5.44), (6.5.41) discussed in Complement 6.5.4. This observation has two consequences. First, it follows that the combined RCM design problem in (6.5.44), (6.5.41) could be solved by an algorithm similar to the one presented below for solving the CCM problem; indeed, this is the case, as is shown in [Li, Stoica, and Wang 2004]. Second, the CCM problem (6.5.70) and the combined RCM problem (6.5.44), (6.5.41) both have two constraints and are more complicated than the RCM problem (6.5.44), which has only one constraint. Hence, the CCM algorithm described below will be (slightly) more involved computationally than the RCM algorithm outlined in Complement 6.5.4. ■

We begin the analysis of the CCM problem in (6.5.70) by deriving a feasible range for the user parameter c. Let S denote the set of vectors h that satisfy both constraints in (6.5.70):

$$S = \{h \mid h^* a = 1 \text{ and } \|h\|^2 \le c\} \tag{6.5.75}$$

By the Cauchy–Schwartz inequality (see Result R12 in Appendix A), we have that

$$1 = \left|h^* a\right|^2 \le \|h\|^2 \|a\|^2 \le cm \implies c \ge \frac{1}{m} \tag{6.5.76}$$

where we also used the fact that (by assumption; see (6.3.11)),

$$\|a\|^2 = m \tag{6.5.77}$$

The inequality in (6.5.76) sets a lower bound on c; otherwise, S is empty. To obtain an upper bound, we can argue as follows: The vector h used in the CM has the norm

$$\|h_{\text{CM}}\|^2 = \frac{a^* R^{-2} a}{\left(a^* R^{-1} a\right)^2} \tag{6.5.78}$$

Because the noise gain of the CM is typically too high, we should like to choose c so that

$$c < \frac{a^* R^{-2} a}{\left(a^* R^{-1} a\right)^2} \tag{6.5.79}$$

Note that, if c does not satisfy (6.5.79), then the CM spatial filter h satisfies both constraints in (6.5.70) and, hence, *is the solution to the CCM problem.* Combining (6.5.76) and (6.5.79) yields the following interval for c:

$$\boxed{c \in \left[\frac{1}{m}, \frac{a^* R^{-2} a}{\left(a^* R^{-1} a\right)^2}\right]} \tag{6.5.80}$$

As with (6.5.53), let

$$R = U \Lambda U^* \tag{6.5.81}$$

be the eigenvalue decomposition (EVD) of R, where $U^* U = UU^* = I$ and

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}; \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \tag{6.5.82}$$

Now,

$$\frac{a^* R^{-2} a}{\left[a^* R^{-1} a\right]^2} \leq \frac{\|a\|^2 / \lambda_m^2}{\left[\|a\|^2 / \lambda_1\right]^2} = \frac{\lambda_1^2}{m \lambda_m^2} \tag{6.5.83}$$

so it follows from (6.5.79) that c also satisfies

$$mc < \frac{\lambda_1^2}{\lambda_m^2} \tag{6.5.84}$$

The above inequality will be useful later on.

Next, let us define the function

$$g(h, \lambda, \mu) = h^* R h + \lambda(\|h\|^2 - c) + \mu(-h^* a - a^* h + 2) \tag{6.5.85}$$

where $\mu \in \mathbf{R}$ is arbitrary and where

$$\lambda > 0 \tag{6.5.86}$$

Remark: We note, in passing, that λ and μ are the so-called Lagrange multipliers and that $g(h, \lambda, \mu)$ is the so-called Lagrangian function associated with the CCM problem in (6.5.70); however, to make the following derivation as self-contained as possible, we will not explicitly use any result from Lagrange-multiplier theory. ■

Evidently, by the definition of $g(h, \lambda, \mu)$, we have that

$$g(h, \lambda, \mu) \leq h^* R h \qquad \text{for any } h \in S \tag{6.5.87}$$

and for any $\mu \in \mathbf{R}$ and $\lambda > 0$. The part of (6.5.85) that depends on h can be written as

$$\begin{aligned} & h^*(R+\lambda I)h - \mu h^* a - \mu a^* h \\ & = \left[h - \mu(R+\lambda I)^{-1}a\right]^* (R+\lambda I)\left[h - \mu(R+\lambda I)^{-1}a\right] \\ & \quad - \mu^2 a^*(R+\lambda I)^{-1}a \end{aligned} \tag{6.5.88}$$

Hence, for fixed λ and μ, the *unconstrained minimizer* of $g(h, \lambda, \mu)$ with respect to h is given by

$$\hat{h}(\lambda, \mu) = \mu(R+\lambda I)^{-1}a \tag{6.5.89}$$

Let us choose μ such that (6.5.89) satisfies the first constraint in (6.5.70):

$$\hat{h}^*(\lambda, \hat{\mu})a = 1 \iff \hat{\mu} = \frac{1}{a^*(R+\lambda I)^{-1}a} \tag{6.5.90}$$

(which is always possible, for $\lambda > 0$). Also, let us choose λ so that (6.5.89) also satisfies the second constraint in (6.5.70) *with equality*—that is,

$$\|\hat{h}(\hat{\lambda}, \hat{\mu})\|^2 = c \iff \frac{a^*(R+\hat{\lambda} I)^{-2}a}{[a^*(R+\hat{\lambda} I)^{-1}a]^2} = c \tag{6.5.91}$$

We will show shortly that this equation has a unique solution $\hat{\lambda} > 0$ for *any* c satisfying (6.5.80). Before doing so, we remark on the following important fact: Inserting (6.5.90) into (6.5.89), we get the diagonally loaded version of the Capon method (see (6.5.68))—that is,

$$\hat{h}(\hat{\lambda}, \hat{\mu}) = \frac{(R+\hat{\lambda} I)^{-1}a}{a^*(R+\hat{\lambda} I)^{-1}a} \tag{6.5.92}$$

Because $\hat{\lambda}$ satisfies (6.5.91), the vector $\hat{h}(\hat{\lambda}, \hat{\mu})$ lies on the boundary of S; hence (see also (6.5.87)),

$$g\left(\hat{h}(\hat{\lambda}, \hat{\mu}), \hat{\lambda}, \hat{\mu}\right) = \hat{h}^*(\hat{\lambda}, \hat{\mu})R\hat{h}(\hat{\lambda}, \hat{\mu}) \leq h^*Rh \quad \text{for any } h \in S \tag{6.5.93}$$

From (6.5.93), we conclude that (6.5.92) *is the (unique) solution to the CCM problem.*

It remains to show that, indeed, equation (6.5.91) has a unique solution $\hat{\lambda} > 0$ under (6.5.80) and also to provide a computationally convenient way of finding $\hat{\lambda}$. Towards that end, we use the EVD of R in (6.5.91) (with the hat on $\hat{\lambda}$ omitted, for notational simplicity) to rewrite (6.5.91) as follows:

$$f(\lambda) = c \tag{6.5.94}$$

where

$$f(\lambda) = \frac{a^*(R+\lambda I)^{-2}a}{\left[a^*(R+\lambda I)^{-1}a\right]^2} = \frac{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]}{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^2} \tag{6.5.95}$$

and where b_k is the kth element of the vector

$$b = U^*a \tag{6.5.96}$$

Differentiation of (6.5.95) with respect to λ yields

$$\begin{aligned} f'(\lambda) = &\left\{ -2\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^2 \right. \\ &\left. +2\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]\right\} \\ &\cdot \frac{1}{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^4} \\ = -2&\left\{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right] - \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]^2\right\} \\ &\cdot \frac{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]}{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^4} \end{aligned} \tag{6.5.97}$$

Making use of the Cauchy–Schwartz inequality once again, we can show that

$$\begin{aligned} \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]^2 &= \left[\sum_{k=1}^{m} \frac{|b_k|}{(\lambda_k+\lambda)^{3/2}} \frac{|b_k|}{(\lambda_k+\lambda)^{1/2}}\right]^2 \\ &< \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right] \end{aligned} \tag{6.5.98}$$

Hence,

$$\begin{aligned} &f'(\lambda) < 0 \text{ for any } \lambda > 0 \\ &\qquad (\text{and } \lambda_k \neq \lambda_p \text{ for at least one pair } k \neq p) \end{aligned} \tag{6.5.99}$$

which means that $f(\lambda)$ is a *monotonically strictly decreasing function* for $\lambda > 0$. Combining this observation with the fact that $f(0) > c$ (see (6.5.79)) shows that, indeed, the equation $f(\lambda) = c$ in (6.5.91) has a *unique solution* for $\lambda > 0$.

For efficiently solving the equation $f(\lambda) = c$, an upper bound on λ would also be useful. Such a bound can be obtained from (6.5.95) as follows: A simple calculation shows that

$$\begin{aligned} c = f(\lambda) &< \frac{\dfrac{\|b\|^2}{(\lambda_m + \lambda)^2}}{\dfrac{\|b\|^4}{(\lambda_1 + \lambda)^2}} = \frac{(\lambda_1 + \lambda)^2}{m(\lambda_m + \lambda)^2} \\ &\Longrightarrow mc(\lambda_m + \lambda)^2 < (\lambda_1 + \lambda)^2 \end{aligned} \tag{6.5.100}$$

where we used the fact that $\|b\|^2 = \|a\|^2 = m$. From (6.5.100), we see that λ must satisfy the inequality

$$\lambda < \frac{\lambda_1 - \sqrt{mc}\lambda_m}{\sqrt{mc} - 1} \triangleq \lambda_U \tag{6.5.101}$$

Note that both the numerator and the denominator in (6.5.101) are positive; see (6.5.76) and (6.5.84).

The derivation of the constrained Capon method is now complete. The following is *a step-by-step summary of the CCM.*

The Constrained Capon Algorithm

Step 1. Compute the eigendecomposition $R = U\Lambda U^*$, and set $b = U^*a$.

Step 2. Solve the equation $f(\lambda) = c$ for λ (using, e.g., a Newton method, along with the fact that there is a unique solution that lies in the interval $(0, \lambda_U)$).

Step 3. Compute the (diagonally loaded) spatial filter vector

$$h = \frac{(R + \lambda I)^{-1}a}{a^*(R + \lambda I)^{-1}a} = \frac{U(\Lambda + \lambda I)^{-1}b}{b^*(\Lambda + \lambda I)^{-1}b}$$

where λ is found in Step 2, and estimate the signal power as h^*Rh.

To conclude this complement, we note that the CCM algorithm is quite similar to the RCM algorithm presented in Complement 6.5.4. The only differences are that the equation for λ

associated with the CCM is slightly more complicated and, more importantly, that it is harder to select the c needed in the CCM (for any DOA of interest) than it is to select ε in the RCM. As we have shown, for CCM one should choose c in the interval (6.5.80). Note that for $c = 1/m$ we get $\lambda \to \infty$ and $h = a/m$, which is the beamforming method. For $c = a^*R^{-2}a/(a^*R^{-1}a)^2$, we obtain $\lambda = 0$ and $h = h_{\text{CM}}$, which is the standard Capon method. Values of c between these two extremes should be chosen in an application-dependent manner.

6.5.6 Spatial Amplitude and Phase Estimation (APES)

As was explained in Section 6.3.2, the Capon method estimates the spatial spectrum by using a spatial filter that passes the signal impinging on the array from direction θ in a distortionless manner and at the same time attenuates signals with DOAs different from θ as much as possible. The Capon method for temporal spectral analysis is based on exactly the same idea (see Section 5.4), as is the temporal APES method described in Complement 5.6.4. In this complement, we will present an extension of APES that can be used for spatial spectral analysis.

Let θ denote a generic DOA, and consider the equation (6.2.19),

$$y(t) = a(\theta)s(t) + e(t), \qquad t = 1, \ldots, N \tag{6.5.102}$$

that describes the array output, $y(t)$, as a function of a signal, $s(t)$, possibly impinging on the array from a DOA equal to θ, and a term, $e(t)$, that includes noise along with any other signals whose DOAs are different from θ. We assume that the array is *uniform and linear*, in which case $a(\theta)$ is given by

$$a(\theta) = \left[1, e^{-i\omega_s}, \ldots, e^{-i(m-1)\omega_s}\right]^T \tag{6.5.103}$$

where m denotes the number of sensors in the array, and $\omega_s = (\omega_c d \sin\theta)/c$ is the spatial frequency (see (6.2.26) and (6.2.27)). As we will explain later, the spatial extension of APES presented in this complement appears to perform well only in the case of ULAs. This is a limitation, but it is not a serious one, because there are techniques that can be used to approximately transform the direction vector of a general array into the direction vector of a fictitious ULA (see, for example, [Doron, Doron, and Weiss 1993]). Such a technique performs a relatively simple DOA-independent linear transformation of the array output snapshots; these linearly transformed snapshots can then be used as the input to the spatial APES method presented here. (See [Abrahamsson, Jakobsson, and Stoica 2004] for details on how to use the spatial APES approach of this complement for arrays that are not uniform and linear.)

Let σ_s^2 denote the power of the signal $s(t)$ in (6.5.102), which is the main parameter we want to estimate; note that the estimated signal power $\hat{\sigma}_s^2$, as a function of θ, provides an estimate of the spatial spectrum. In this complement, we assume that $\{s(t)\}_{t=1}^N$ is an unknown *deterministic* sequence; hence, we define σ_s^2 as

$$\sigma_s^2 = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} |s(t)|^2 \tag{6.5.104}$$

An important difference between equation (6.5.102) and its temporal counterpart (see, e.g., equation (5.6.80) in Complement 5.6.6) is that, in (6.5.102), the signal $s(t)$ is *completely unknown*, whereas, in the temporal case, we have $s(t) = \beta e^{i\omega t}$ and only the amplitude is unknown. Because of this difference, the use of the APES principle for spatial spectral estimation is somewhat different from its use for temporal spectral estimation.

Remark: We remind the reader that $\{s(t)\}_{t=1}^{N}$ is assumed to be an unknown deterministic sequence here. The case in which $\{s(t)\}$ is assumed to be stochastic is considered in Complement 6.5.3. Interestingly, application of the APES principle in the stochastic signal case leads to the (standard) Capon method! ■

Let $\bar{m} < m$ be an integer, and define the following two vectors:

$$\bar{a}_k = \left[e^{-i(k-1)\omega_s}, e^{-ik\omega_s}, \ldots, e^{-i(k+\bar{m}-2)\omega_s}\right]^T \qquad (\bar{m} \times 1) \qquad (6.5.105)$$

$$\bar{y}_k(t) = \left[y_k(t), y_{k+1}(t), \ldots, y_{k+\bar{m}-1}(t)\right]^T \qquad (\bar{m} \times 1) \qquad (6.5.106)$$

for $k = 1, \ldots, L$, with

$$L = m - \bar{m} + 1 \qquad (6.5.107)$$

In (6.5.106), $y_k(t)$ denotes the kth element of $y(t)$; also, we omit the dependence of $\bar{a}_k$ on θ to simplify notation. The choice of the user parameter $\bar{m}$ will be discussed later.

The assumed ULA structure means that the direction subvectors $\{\bar{a}_k\}$ satisfy the following relations:

$$\bar{a}_k = e^{-i(k-1)\omega_s}\bar{a}_1, \qquad k = 2, \ldots, L \qquad (6.5.108)$$

Consequently, $\bar{y}_k(t)$ can be written (see (6.5.102)) as

$$\bar{y}_k(t) = \bar{a}_k s(t) + \bar{e}_k(t) = e^{-i(k-1)\omega_s}\bar{a}_1 s(t) + \bar{e}_k(t) \qquad (6.5.109)$$

where $\bar{e}_k(t)$ is a noise vector defined similarly to $\bar{y}_k(t)$. Let h denote the $(\bar{m} \times 1)$ coefficient vector of a spatial filter that is applied to $\{e^{i(k-1)\omega_s}\bar{y}_k(t)\}_{k=1}^{L}$. Then it follows from (6.5.109) that h passes the signal $s(t)$ in each of these data sets in a distortionless manner if and only if

$$h^*\bar{a}_1 = 1 \qquad (6.5.110)$$

Using the preceding observations along with the APES principle presented in Complement 5.6.4, we can determine both the spatial filter h and an estimate of the complex-valued sequence $\{s(t)\}_{t=1}^{N}$ (we estimate both amplitude and phase—recall that APES stands for Amplitude and Phase EStimation) by solving the following linearly constrained least-squares (LS) problem:

$$\min_{h;\{s(t)\}} \sum_{t=1}^{N}\sum_{k=1}^{L}\left|h^*\bar{y}_k(t)e^{i(k-1)\omega_s} - s(t)\right|^2 \quad \text{subject to: } h^*\bar{a}_1 = 1 \qquad (6.5.111)$$

The quadratic criterion in (6.5.111) expresses our desire to make the outputs of the spatial filter, $\{h^* \bar{y}_k(t) e^{i(k-1)\omega_s}\}_{k=1}^L$, resemble a signal $s(t)$ (that is independent of k) as much as possible, in a least-squares sense. Said another way, this LS criterion expresses our goal to make the filter h attenuate any signal in $\{\bar{y}_k(t) e^{i(k-1)\omega_s}\}_{k=1}^L$, whose DOA is different from θ, as much as possible. The linear constraint in (6.5.111) forces the spatial filter h to pass the signal $s(t)$ undistorted.

To derive a solution to (6.5.111), let

$$g(t) = \frac{1}{L} \sum_{k=1}^{L} \bar{y}_k(t) e^{i(k-1)\omega_s} \tag{6.5.112}$$

and observe that

$$\begin{aligned}
&\frac{1}{L} \sum_{k=1}^{L} \left| h^* \bar{y}_k(t) e^{i(k-1)\omega_s} - s(t) \right|^2 \\
&\quad = |s(t)|^2 + h^* \left[\frac{1}{L} \sum_{k=1}^{L} \bar{y}_k(t) \bar{y}_k^*(t) \right] h - h^* g(t) s^*(t) - g^*(t) h s(t) \\
&\quad = h^* \left[\frac{1}{L} \sum_{k=1}^{L} \bar{y}_k(t) \bar{y}_k^*(t) \right] h - h^* g(t) g^*(t) h + \left| s(t) - h^* g(t) \right|^2
\end{aligned} \tag{6.5.113}$$

Hence, the sequence $\{s(t)\}$ that minimizes (6.5.111), for fixed h, is given by

$$\boxed{\hat{s}(t) = h^* g(t)} \tag{6.5.114}$$

Inserting (6.5.114) into (6.5.111) (see also (6.5.113)), we obtain the reduced problem

$$\min_h h^* \hat{Q} h \qquad \text{subject to: } h^* \bar{a}_1 = 1 \tag{6.5.115}$$

where

$$\begin{aligned}
\hat{Q} &= \hat{R} - \hat{G} \\
\hat{R} &= \frac{1}{N} \sum_{t=1}^{N} \frac{1}{L} \sum_{k=1}^{L} \bar{y}_k(t) \bar{y}_k^*(t) \\
\hat{G} &= \frac{1}{N} \sum_{t=1}^{N} g(t) g^*(t)
\end{aligned} \tag{6.5.116}$$

The solution to the quadratic problem with linear constraints in (6.5.115) can be obtained by using Result R35 in Appendix A:

$$\boxed{\hat{h} = \frac{\hat{Q}^{-1}\bar{a}_1}{\bar{a}_1^*\hat{Q}^{-1}\bar{a}_1}} \tag{6.5.117}$$

Using (6.5.117) in (6.5.114), we can obtain both an estimate of the signal sequence, which may be of interest in some applications, and an estimate of the signal power:

$$\boxed{\hat{\sigma}_s^2 = \frac{1}{N}\sum_{t=1}^{N}\left|\hat{s}(t)\right|^2 = \hat{h}^*\hat{G}\hat{h}} \tag{6.5.118}$$

This equation, as a function of DOA θ, provides an estimate of the spatial spectrum.

The matrix $\hat{Q}$ in (6.5.116) can be rewritten in the following form:

$$\hat{Q} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{L}\sum_{k=1}^{L}\left[e^{i(k-1)\omega_s}\bar{y}_k(t) - g(t)\right]\left[e^{i(k-1)\omega_s}\bar{y}_k(t) - g(t)\right]^* \tag{6.5.119}$$

It follows from (6.5.119) that $\hat{Q}$ is always positive semidefinite. For $L = 1$ (or, equivalently, $\bar{m} = m$), we have $\hat{Q} = 0$ because $g(t) = \bar{y}_1(t)$ for $t = 1, \ldots, N$. Thus, for $L = 1$, (6.5.117) is not valid. This is expected: indeed, for $L = 1$ we can make (6.5.111) equal to zero, for *any* h, by choosing $\hat{s}(t) = h^*\bar{y}_1(t)$; consequently, the problem of minimizing (6.5.111) with respect to $(h; \{s(t)\}_{t=1}^N)$ is underdetermined for $L = 1$, and, hence, an infinite number of solutions exist. To prevent this from happening, we should choose $L \geq 2$ (or, equivalently, $\bar{m} \leq m - 1$). For $L \geq 2$, the $(\bar{m} \times \bar{m})$ matrix $\hat{Q}$ is a sum of NL outer products; if $NL \geq \bar{m}$, which is a weak condition, $\hat{Q}$ is almost surely strictly positive definite and hence nonsingular.

From a performance point of view, it turns out that a good choice of $\bar{m}$ is its maximum possible value:

$$\bar{m} = m - 1 \qquad \Longleftrightarrow \qquad L = 2 \tag{6.5.120}$$

A numerical study of performance, reported in [GINI AND LOMBARDINI 2002], supports this choice of $\bar{m}$ and also suggests that the spatial APES method can outperform the Capon method both in spatial-spectrum estimation and in DOA-estimation applications. The APES spatial filter is, however, more difficult to compute than is the Capon spatial filter, as a result of the dependence of $\hat{Q}$ in (6.5.117) on the DOA.

In the remainder of this complement, we will explain why the APES method may be expected to outperform the Capon method. In doing so, we assume that $\bar{m} = m - 1$ (and thus $L = 2$), as in (6.5.120). Intuitively, this choice of $\bar{m}$ provides the APES filter with the maximum possible number of degrees of freedom; hence, it makes sense that it should lead to better resolution and interference-rejection capability than would smaller values of $\bar{m}$. For $L = 2$, we have

$$g(t) = \frac{1}{2}[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)] \tag{6.5.121}$$

and, hence,

$$\begin{aligned}\hat{Q} &= \frac{1}{2N}\sum_{t=1}^{N}\left\{\frac{1}{4}[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)][\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)]^*\right.\\ &\qquad \left. + \frac{1}{4}[e^{i\omega_s}\bar{y}_2(t) - \bar{y}_1(t)][e^{i\omega_s}\bar{y}_2(t) - \bar{y}_1(t)]^*\right\}\\ &= \frac{1}{4N}\sum_{t=1}^{N}[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)][\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)]^* \end{aligned} \tag{6.5.122}$$

It follows that the APES spatial filter is the solution to the problem (see (6.5.115))

$$\min_h \sum_{t=1}^{N}\left|h^*\left[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)\right]\right|^2 \quad \text{subject to: } h^*\bar{a}_1 = 1 \tag{6.5.123}$$

and that the APES signal estimate is given (see (6.5.114)) by

$$\hat{s}(t) = \frac{1}{2}h^*\left[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)\right] \tag{6.5.124}$$

On the other hand, the Capon spatial filter is obtained as the solution to the problem

$$\min_h \sum_{t=1}^{N}\left|h^*y(t)\right|^2 \quad \text{subject to: } h^*a = 1 \tag{6.5.125}$$

and the Capon signal estimate is given by

$$\hat{s}(t) = h^*y(t) \tag{6.5.126}$$

To explain the main differences between the APES and Capon approaches, let us assume that, in addition to the signal of interest (SOI) $s(t)$ impinging on the array from the DOA under consideration θ, there is an interference signal $i(t)$ that impinges on the array from another DOA, denoted θ_i. We consider the situation in which only one interference signal is present, to simplify the discussion, but the case of multiple interference signals can be treated similarly. The array output vector in (6.5.102) and the subvectors in (6.5.109) become

$$y(t) = a(\theta)s(t) + b(\theta_i)i(t) + e(t) \tag{6.5.127}$$

$$\bar{y}_1(t) = \bar{a}_1(\theta)s(t) + \bar{b}_1(\theta_i)i(t) + \bar{e}_1(t) \tag{6.5.128}$$

$$\bar{y}_2(t) = \bar{a}_2(\theta)s(t) + \bar{b}_2(\theta_i)i(t) + \bar{e}_2(t) \tag{6.5.129}$$

where the quantities b, $\bar{b}_1$, and $\bar{b}_2$ are defined similarly to a, $\bar{a}_1$, and $\bar{a}_2$. We have shown the dependence of the various quantities on θ and θ_i in equations (6.5.127)–(6.5.129), but we will drop the DOA dependence in the remainder of the derivation, to simplify notation.

For the previous scenario, the Capon method is known to have poor performance in either of the following two situations:

(i) The SOI steering vector is known imprecisely—for example, as a result of pointing or calibration errors;
(ii) The SOI is highly correlated or coherent with the interference, which happens in multipath-propagation or smart-jamming scenarios.

To explain the difficulty of the Capon method in case (i), let us assume that the true steering vector of the SOI is $a_0 \neq a$. Then, by design, the Capon filter will be such that $|h^*a_0| \simeq 0$ (where $\simeq 0$ denotes a "small" value). Therefore, the SOI, whose steering vector is different from the assumed vector a, is treated as an interference signal and is attenuated or cancelled. As a consequence, the power of the SOI will be significantly underestimated, unless special measures are taken to make the Capon method robust against steering-vector errors (see Complements 6.5.4 and 6.5.5.)

The performance degradation of the Capon method in case (ii) is also easy to understand. Assume that the interference is coherent with the SOI and, hence, that $i(t) = \rho s(t)$ for some nonzero constant ρ. Then (6.5.127) can be rewritten as

$$y(t) = (a + \rho b)s(t) + e(t) \tag{6.5.130}$$

which shows that the SOI steering vector is given by $(a + \rho b)$ in lieu of the assumed vector a. Consequently, the Capon filter will by design be such that $|h^*(a + \rho b)| \simeq 0$, and therefore the SOI will be attenuated or cancelled in the filter output $h^*y(t)$, as in case (i). In fact, case (ii) can be considered as an extreme example of case (i), in which the SOI steering vector errors can be significant. Modifying the Capon method to work well in the case of coherent multipath signals is thus a more difficult problem than modifying it to be robust to small steering-vector errors.

Next, let us consider the APES method in case (ii). From (6.5.128) and (6.5.129), along with (6.5.108), we get

$$\begin{aligned}
&\left[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)\right] \\
&\quad = \left(\bar{a}_1 - e^{i\omega_s}\bar{a}_2\right) s(t) + \left(\bar{b}_1 - e^{i\omega_s}\bar{b}_2\right) i(t) + \left[\bar{e}_1(t) - e^{i\omega_s}\bar{e}_2(t)\right] \\
&\quad = \left[1 - e^{i(\omega_s - \omega_i)}\right] \bar{b}_1 i(t) + \left[\bar{e}_1(t) - e^{i\omega_s}\bar{e}_2(t)\right]
\end{aligned} \tag{6.5.131}$$

and

$$\begin{aligned}
&\frac{1}{2}\left[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)\right] \\
&\quad = \frac{1}{2}\left(\bar{a}_1 + e^{i\omega_s}\bar{a}_2\right) s(t) + \frac{1}{2}\left(\bar{b}_1 + e^{i\omega_s}\bar{b}_2\right) i(t) + \frac{1}{2}\left[\bar{e}_1(t) + e^{i\omega_s}\bar{e}_2(t)\right] \\
&\quad = \bar{a}_1 s(t) + \frac{1}{2}\left[1 + e^{i(\omega_s - \omega_i)}\right] \bar{b}_1 i(t) + \frac{1}{2}\left[\bar{e}_1(t) + e^{i\omega_s}\bar{e}_2(t)\right]
\end{aligned} \tag{6.5.132}$$

where $\omega_i = (\omega_c d \sin\theta_i)/c$ denotes the spatial frequency of the interference. It follows from (6.5.131) and the design criterion in (6.5.123) that the APES spatial filter will be such that

$$\left|1 - e^{i(\omega_s - \omega_i)}\right| \cdot \left|h^* \bar{b}_1\right| \simeq 0 \tag{6.5.133}$$

Hence, because the SOI is absent from the data vector in (6.5.131), the APES filter is able to cancel the interference only, despite the fact that the interference and the SOI are coherent. This interference-rejection property of the APES filter (i.e., $|h^*\bar{b}_1| \simeq 0$) is precisely what is needed when estimating the SOI from the data in (6.5.132).

To summarize, the APES method circumvents the problem in case (ii) by implicitly eliminating the signal from the data that is used to derive the spatial filter. However, if there is more than one coherent interference in the observed data, then APES also breaks down, in much the way that the Capon method does. The reason is that the vector multiplying $i(t)$ in (6.5.131) is no longer proportional to the vector multiplying $i(t)$ in (6.5.132); hence, a filter h that, by design, cancels the interference $i(t)$ in (6.5.131) is not guaranteed to have the desirable effect of cancelling $i(t)$ in (6.5.132); the details are left to the interested reader.

Remark: An argument similar to the one above explains why APES will not work well for non-ULA array geometries, in spite of the fact that it can be extended to such geometries in a relatively straightforward manner. Specifically, for non-ULA geometries, the steering vectors of the interference terms in the data sets used to obtain h and to estimate $s(t)$, respectively, are not proportional to one another. As a consequence, the design objective does not provide the APES filter with the desired capability of attenuating the interference terms in the data that is used to estimate $\{s(t)\}$. ■

Next consider the APES method in case (i). To simplify the discussion, let us assume that there are no calibration errors, but only a pointing error, so that the true spatial frequency of the SOI is $\omega_s^0 \neq \omega_s$. Then equation (6.5.131) becomes

$$\begin{aligned}\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t) = \left[1 - e^{i(\omega_s - \omega_s^0)}\right]\bar{a}_1^0 s(t) + \left[1 - e^{i(\omega_s - \omega_i)}\right]\bar{b}_1 i(t) \\ + \left[\bar{e}_1(t) - e^{i\omega_s}\bar{e}_2(t)\right]\end{aligned} \tag{6.5.134}$$

It follows that, in case (i), the APES spatial filter tends to cancel the SOI, in addition to cancelling the interference. However, the pointing errors are usually quite small; therefore, the residual term of $s(t)$ in (6.5.134) is small as well. Hence, the SOI might well pass through the APES filter (i.e., $|h^*\bar{a}_1^0|$ may be reasonably close to $|h^*\bar{a}_1| = 1$), because the filter uses most of its degrees of freedom to cancel the much stronger interference term in (6.5.134). As a consequence, APES is less sensitive to steering-vector errors than is the Capon method.

The previous discussion also explains why APES can provide *better power estimates* than the Capon method, even in "ideal" cases in which there are no multipath signals that are coherent with the SOI and no steering-vector errors, but the number of snapshots N is not very large. Indeed, as argued in Complement 6.5.5, the finite-sample effects associated with practical values of N can be viewed as inducing both correlation among the signals and steering-vector errors, to which the APES method is less sensitive than the Capon method, as has been explained.

We also note that the power of the elements of the noise vector in the data in (6.5.131) that is used to derive the APES filter is larger than the power of the noise elements in the raw data $y(t)$ that is used to compute the Capon filter. Somewhat counterintuitively, this is another potential advantage of the APES method over the Capon method. Indeed, the increased noise power in the data used by APES has a regularizing effect on the APES filter, which keeps the filter noise gain down, whereas the Capon filter is known to have a relatively large noise gain that can have a detrimental effect on signal-power estimation. (See Complement 6.5.5.)

On the downside, APES has been found to have a *slightly lower resolution* than the Capon method. (See, e.g., [Jakobsson and Stoica 2000].) Our previous discussion also provides a simple explanation of this result: when the interference and the SOI are closely spaced (i.e., when $\omega_s \simeq \omega_i$), the first factor in (6.5.133) becomes rather small and so might allow the second factor to increase somewhat. This explains how the beamwidth of the APES spatial filter could be larger than that of the Capon filter and, hence, why APES might have a slightly lower resolution.

6.5.7 The CLEAN Algorithm

The CLEAN algorithm is a *semiparametric method* that can be used for spatial spectral estimation. As we will see, this algorithm can be introduced in a nonparametric fashion (see [Högbom 1974]), yet its performance depends heavily on an implicit parametric assumption about the structure of the spatial covariance matrix; thus, CLEAN lies in between the class of nonparametric and parametric approaches, and it can be called a semiparametric approach.

There is a significant literature about CLEAN and its many applications in diverse areas, including array signal processing, image processing, and astronomy (see, for example, [Cornwell and Bridle 1996] and its references.) Our discussion of CLEAN will focus on its application to spatial spectral analysis and DOA estimation.

First, we present an *intuitive motivation* of CLEAN. Consider the beamforming spatial spectral estimate in (6.3.18), namely,

$$\hat{\phi}_1(\theta) = a^*(\theta)\hat{R}a(\theta) \tag{6.5.135}$$

where $a(\theta)$ and $\hat{R}$ are defined as in Section 6.3.1. Let

$$\hat{\theta}_1 = \arg\max_{\theta} \hat{\phi}_1(\theta) \tag{6.5.136}$$

$$\hat{\sigma}_1^2 = \frac{1}{m^2}\hat{\phi}_1(\hat{\theta}_1) \tag{6.5.137}$$

In words, $\hat{\sigma}_1^2$ is the scaled height of the highest peak of $\hat{\phi}_1(\theta)$, and $\hat{\theta}_1$ is its corresponding DOA (see (6.3.16) and (6.3.18)). As we know, the beamforming method suffers from resolution and leakage problems. However, the dominant peak of the beamforming spectrum, $\hat{\phi}_1(\theta)$, is likely to indicate that there is a source, or possibly several closely-spaced sources, at or in the vicinity of $\hat{\theta}_1$. The covariance matrix of the part of the array output due to a source signal with DOA equal to $\hat{\theta}_1$ and power equal to $\hat{\sigma}_1^2$ is given (see, e.g., (6.2.19)) by

$$\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1) \tag{6.5.138}$$

Consequently, the expected term in $\hat{\phi}_1(\theta)$ due to (6.5.138) is

$$\hat{\sigma}_1^2 \left| a^*(\theta) a(\hat{\theta}_1) \right|^2 \tag{6.5.139}$$

We *partly* eliminate the term (6.5.139) from $\hat{\phi}_1(\theta)$ and thereby define a new spectrum

$$\hat{\phi}_2(\theta) = \hat{\phi}_1(\theta) - \rho \hat{\sigma}_1^2 \left| a^*(\theta) a(\hat{\theta}_1) \right|^2 \tag{6.5.140}$$

where ρ is a user parameter that satisfies

$$\rho \in (0, 1] \tag{6.5.141}$$

The reason for using a value of $\rho < 1$ in (6.5.140) can be explained as follows:

(a) The assumption that there is a source with parameters $(\hat{\sigma}_1^2, \hat{\theta}_1)$ corresponding to the maximum peak of the beamforming spectrum, which led to (6.5.140), is not necessarily true. For example, there could be *several sources* clustered around $\hat{\theta}_1$ that were not resolved by the beamforming method. Subtracting only a (small) part of the beamforming response to a source signal with parameters $(\hat{\sigma}_1^2, \hat{\theta}_1)$ leaves "some power" at and around $\hat{\theta}_1$. Hence, the algorithm will likely return to this DOA region of the beamforming spectrum in future iterations, when it could have a better chance to resolve the power around $\hat{\theta}_1$ into its true constituent components.

(b) Even if there is indeed a single source at or close to $\hat{\theta}_1$, the estimation of its parameters could be affected by leakage from other sources; this leakage will be particularly strong when the source signal in question is correlated with other source signals. In such a case, (6.5.139) is a *poor estimate* of the contribution of the source in question to the beamforming spectrum. By subtracting only a part of (6.5.139) from $\hat{\phi}_1(\theta)$, we give the algorithm a chance to improve the parameter estimates of the source at or close to $\hat{\theta}_1$ in future iterations.

(c) In both situations (a) and (b), and possibly in other cases as well, in which (6.5.139) is a poor approximation of the part of the beamforming spectrum that is due to the source(s) at or around $\hat{\theta}_1$, subtracting (6.5.139) from $\hat{\phi}_1(\theta)$ fully (i.e., using $\rho = 1$) might yield a *spatial spectrum that takes on negative values* at some DOAs (as it never should). Using $\rho < 1$ in (6.5.140) reduces the likelihood that this undesirable event will happen too early in the iterative process of the CLEAN algorithm (as we discuss later).

The calculation of $\hat{\phi}_2(\theta)$, as in (6.5.140), completes the first iteration of CLEAN. In the second iteration, we proceed similarly, but using $\hat{\phi}_2(\theta)$ instead of $\hat{\phi}_1(\theta)$. Hence, we let

$$\hat{\theta}_2 = \arg\max_{\theta} \hat{\phi}_2(\theta) \tag{6.5.142}$$

$$\hat{\sigma}_2^2 = \frac{1}{m^2} \hat{\phi}_2(\hat{\theta}_2) \tag{6.5.143}$$

and

$$\hat{\phi}_3(\theta) = \hat{\phi}_2(\theta) - \rho\hat{\sigma}_2^2 \left| a^*(\theta)a(\hat{\theta}_2) \right|^2 \tag{6.5.144}$$

Continuing the iterations in the same manner yields the CLEAN algorithm, a compact description of which is as follows:

The CLEAN Algorithm

```
Initialization:   $\hat{\phi}_1(\theta) = a^*(\theta)\hat{R}a(\theta)$
For $k = 1, 2, \ldots$ do:
    $\hat{\theta}_k = \arg\max_{\theta} \hat{\phi}_k(\theta)$
    $\hat{\sigma}_k^2 = \frac{1}{m^2}\hat{\phi}_k(\hat{\theta}_k)$
    $\hat{\phi}_{k+1}(\theta) = \hat{\phi}_k(\theta) - \rho\hat{\sigma}_k^2 \left| a^*(\theta)a(\hat{\theta}_k) \right|^2$
```

We continue the iterative process in the CLEAN algorithm until either we complete a prespecified number of iterations or $\hat{\phi}_k(\theta)$, for some k, has become (too) negative at some DOAs (see, for example, [Högbom 1974; Cornwell and Bridle 1996]).

Regarding the choice of ρ in the CLEAN algorithm, there are no clear guidelines about how this choice should be made to enhance the performance of the CLEAN algorithm in a given application; $\rho \in [0.1, 0.25]$ is usually recommended. (See, e.g., [Högbom 1974; Cornwell and Bridle 1996; Schwarz 1978b].) We will make further comments on the choice of ρ later in this complement.

In the CLEAN literature, the beamforming spectral estimate $\hat{\phi}_1(\theta)$ that forms the starting point of CLEAN is called the "dirty" spectrum, in reference to its mainlobe smearing and sidelobe leakage problems. The discrete spatial spectral estimate $\{\rho\hat{\sigma}_k^2, \hat{\theta}_k\}_{k=1,2,\ldots}$ provided by the algorithm (or a suitably smoothed version of it) is called the "clean" spectrum. The iterative process that yields the "clean" spectrum is, then, called the CLEAN algorithm.

It is interesting to observe that the foregoing derivation of CLEAN is not based on a parametric model of the array output or of its covariance matrix, of the type considered in (6.2.21) or (6.4.3). More precisely, we have not made any assumption that there is a finite number of point-source signals impinging on the array, nor that the noise is spatially white. However, we have used the assumption that the covariance matrix due to a source signal has the form in (6.5.138), which cannot be true unless *the signals impinging on the array are uncorrelated with one another*. CLEAN is known to have poor performance if this parametric assumption does not hold. Hence, CLEAN is a combined nonparametric-parametric approach, which we call *semiparametric* for short.

Next, we present a *more formal derivation* of the CLEAN algorithm. Consider the following semiparametric model of the array output covariance matrix:

$$R = \sigma_1^2 a(\theta_1)a^*(\theta_1) + \sigma_2^2 a(\theta_2)a^*(\theta_2) + \cdots \tag{6.5.145}$$

As implied by the previous discussion, this is the covariance model assumed by CLEAN. Let us fit (6.5.145) to the sample covariance matrix $\hat{R}$ in a least-squares sense:

$$\min_{\{\sigma_k^2,\theta_k\}} \left\| \hat{R} - \sigma_1^2 a(\theta_1)a^*(\theta_1) - \sigma_2^2 a(\theta_2)a^*(\theta_2) - \cdots \right\|^2 \tag{6.5.146}$$

We will show that *CLEAN is a sequential algorithm for approximately minimizing the preceding LS covariance-fitting criterion.*

We begin by assuming that the initial estimates of $\sigma_2^2, \sigma_3^2, \ldots$ are equal to zero (in which case $\theta_2, \theta_3, \ldots$ are immaterial). Consequently, we obtain an estimate of the pair (σ_1^2, θ_1) by minimizing (6.5.146) with $\sigma_2^2 = \sigma_3^2 = \cdots = 0$:

$$\min_{\sigma_1^2,\theta_1} \left\| \hat{R} - \sigma_1^2 a(\theta_1)a^*(\theta_1) \right\|^2 \tag{6.5.147}$$

As shown in Complement 6.5.3, the solution to (6.5.147) is given by

$$\hat{\theta}_1 = \arg\max_{\theta} \hat{\phi}_1(\theta); \qquad \hat{\sigma}_1^2 = \frac{1}{m^2}\hat{\phi}_1(\hat{\theta}_1) \tag{6.5.148}$$

where $\hat{\phi}_1(\theta)$ is as defined previously. We reduce this power estimate by using $\rho\hat{\sigma}_1^2$ in lieu of $\hat{\sigma}_1^2$. The reasons for this reduction are discussed in points (a)–(c); in particular, we would like the residual covariance matrix $\hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1)$ to be positive definite. We will discuss this aspect in more detail after completing the derivation of CLEAN.

Next, we obtain an estimate of the pair (σ_2^2, θ_2) by minimizing (6.5.146) with $\sigma_1^2 = \rho\hat{\sigma}_1^2$, $\theta_1 = \hat{\theta}_1$ and $\sigma_3^2 = \sigma_4^2 = \cdots = 0$:

$$\min_{\sigma_2^2,\theta_2} \left\| \hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1) - \sigma_2^2 a(\theta_2)a^*(\theta_2) \right\|^2 \tag{6.5.149}$$

The solution to (6.5.149) can be shown to be (as in solving (6.5.147))

$$\hat{\theta}_2 = \arg\max_{\theta} \hat{\phi}_2(\theta); \qquad \hat{\sigma}_2^2 = \frac{1}{m^2}\hat{\phi}_2(\hat{\theta}_2) \tag{6.5.150}$$

where

$$\begin{aligned} \hat{\phi}_2(\theta) &= a^*(\theta)\left[\hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1)\right]a(\theta) \\ &= \hat{\phi}_1(\theta) - \rho\hat{\sigma}_1^2 \left|a^*(\theta)a(\hat{\theta}_1)\right|^2 \end{aligned} \tag{6.5.151}$$

Observe that (6.5.148) and (6.5.150) coincide with (6.5.136)–(6.5.137) and (6.5.142)–(6.5.143). Evidently, continuing the previous iterative process, for which (6.5.148) and (6.5.150) are the first two steps, leads to the CLEAN algorithm on page 328.

The foregoing derivation of CLEAN sheds some light on the properties of this algorithm. First, note that *the LS covariance-fitting criterion in (6.5.146) is decreased at each iteration of CLEAN*. For instance, consider the first iteration. A straightforward calculation shows that

$$\begin{aligned}
&\left\| \hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1) \right\|^2 \\
&\qquad = \|\hat{R}\|^2 - 2\rho\hat{\sigma}_1^2 a^*(\hat{\theta}_1)\hat{R}a(\hat{\theta}_1) + m^2\rho^2\hat{\sigma}_1^4 \\
&\qquad = \|\hat{R}\|^2 - \rho(2-\rho)m^2\hat{\sigma}_1^4 \qquad (6.5.152)
\end{aligned}$$

Clearly, (6.5.152) is less than $\|\hat{R}\|^2$ for any $\rho \in (0, 2)$, and the maximum decrease occurs for $\rho = 1$ (as expected). A similar calculation shows that the criterion in (6.5.146) monotonically decreases as we continue the iterative process, for any $\rho \in (0, 2)$, and that at each iteration the maximum decrease occurs for $\rho = 1$. As a consequence, we might think of choosing $\rho = 1$, but this is not advisable. The reason is that our goal is not only to decrease the fitting criterion (6.5.146) as much and as fast as possible, but also to ensure that the residual covariance matrices

$$\hat{R}_{k+1} = \hat{R}_k - \rho\hat{\sigma}_k^2 a(\hat{\theta}_k)a^*(\hat{\theta}_k); \qquad \hat{R}_1 = \hat{R} \qquad (6.5.153)$$

remain positive definite for $k = 1, 2, \ldots$; otherwise, fitting $\sigma_{k+1}^2 a(\theta_{k+1})a^*(\theta_{k+1})$ to $\hat{R}_{k+1}$ would make little statistical sense. By a calculation similar to that in equation (6.5.33) of Complement 6.5.3, it can be shown that the condition $\hat{R}_{k+1} > 0$ is equivalent to

$$\rho < \frac{1}{\hat{\sigma}_k^2 a^*(\hat{\theta}_k)\hat{R}_k^{-1}a(\hat{\theta}_k)} \qquad (6.5.154)$$

Note that the right-hand side of (6.5.154) is bounded above by 1, because, by the Cauchy–Schwartz inequality,

$$\begin{aligned}
\hat{\sigma}_k^2 a^*(\hat{\theta}_k)\hat{R}_k^{-1}a(\hat{\theta}_k) &= \frac{1}{m^2}\left[a^*(\hat{\theta}_k)\hat{R}_k a(\hat{\theta}_k)\right]\left[a^*(\hat{\theta}_k)\hat{R}_k^{-1}a(\hat{\theta}_k)\right] \\
&= \frac{1}{m^2}\left\|\hat{R}_k^{1/2}a(\hat{\theta}_k)\right\|^2 \left\|\hat{R}_k^{-1/2}a(\hat{\theta}_k)\right\|^2 \\
&\geq \frac{1}{m^2}\left|a^*(\hat{\theta}_k)\hat{R}_k^{1/2}\hat{R}_k^{-1/2}a(\hat{\theta}_k)\right|^2 \\
&= \frac{1}{m^2}\left|a^*(\hat{\theta}_k)a(\hat{\theta}_k)\right|^2 = 1
\end{aligned}$$

Also note that, depending on the scenario under consideration, satisfaction of the inequality in (6.5.154) for $k = 1, 2, \ldots$ could require choosing a value for ρ much less than 1. To summarize, the preceding discussion has provided a *a precise argument for choosing* $\rho < 1$ (or even $\rho \ll 1$) in the CLEAN algorithm.

The LS covariance-fitting derivation of CLEAN also makes *the semiparametric nature of CLEAN* more transparent. Specifically, the discussion has shown that CLEAN fits the semiparametric covariance model in (6.5.145) to the sample covariance matrix $\hat{R}$.

Finally, note that, although there is a significant literature on CLEAN, its statistical properties are not well understood; in fact, other than the preliminary study of CLEAN reported in [SCHWARZ 1978B], there appear to be very few statistical studies in the literature. The derivation of CLEAN based on the LS covariance-fitting criterion in (6.5.146) could also be useful in *understanding the statistical properties of CLEAN*. However, we will not attempt to provide a statistical analysis of CLEAN in this complement.

6.5.8 Unstructured and Persymmetric ML Estimates of the Covariance Matrix

Let $\{y(t)\}_{t=1,2,\dots}$ be a sequence of independent and identically distributed (i.i.d.) $m \times 1$ random vectors with mean zero and covariance matrix R. The array output given by equation (6.2.21) is an example of such a sequence, under the assumption that the signal $s(t)$ and the noise $e(t)$ in (6.2.21) are temporally white. Furthermore, let $y(t)$ be circularly Gaussian distributed (see Section B.3 in Appendix B), in which case its probability density function is given by

$$p\big(y(t)\big) = \frac{1}{\pi^m |R|} e^{-y^*(t)R^{-1}y(t)} \tag{6.5.155}$$

Assume that N observations of $\{y(t)\}$ are available:

$$\{y(1), \dots, y(N)\} \tag{6.5.156}$$

Owing to the i.i.d. assumption made on the sequence $\{y(t)\}_{t=1,2,\dots}$, the probability density function of the sample in (6.5.156) is given by

$$\begin{aligned} p\big(y(1), \dots, y(N)\big) &= \prod_{t=1}^{N} p\big(y(t)\big) \\ &= \frac{1}{\pi^{mN} |R|^N} e^{-\sum_{t=1}^{N} y^*(t)R^{-1}y(t)} \end{aligned} \tag{6.5.157}$$

The maximum likelihood (ML) estimate of the covariance matrix R, based on the sample in (6.5.156), is given by the maximizer of the likelihood function in (6.5.157) (see Section B.1 in Appendix B) or, equivalently, by the minimizer of the negative log-likelihood function:

$$-\ln p\big(y(1), \dots, y(N)\big) = mN \ln(\pi) + N \ln |R| + \sum_{t=1}^{N} y^*(t)R^{-1}y(t) \tag{6.5.158}$$

The part of (6.5.158) that depends on R is given (after multiplication by $\frac{1}{N}$) by

$$\ln|R| + \frac{1}{N}\sum_{t=1}^{N} y^*(t)R^{-1}y(t) = \ln|R| + \operatorname{tr}\left(R^{-1}\hat{R}\right) \tag{6.5.159}$$

where

$$\hat{R} = \frac{1}{N}\sum_{t=1}^{N} y(t)y^*(t) \qquad (m \times m) \tag{6.5.160}$$

In this complement, we discuss the minimization of (6.5.159) with respect to R, which yields the ML estimate of R, under either of the following two assumptions:

A: R has no assumed structure

or

B: R is persymmetric.

As explained in Section 4.8, R is persymmetric (or centrosymmetric) if and only if

$$JR^TJ = R \quad \Longleftrightarrow \quad R = \frac{1}{2}\left(R + JR^TJ\right) \tag{6.5.161}$$

where J is the so-called reversal matrix defined in (4.8.4).

Remark: If $y(t)$ is the output of an array that is uniform and linear and the source signals are uncorrelated with one another, then the covariance matrix R is Toeplitz, and, hence, persymmetric. ■

We will show that the unstructured ML estimate of R, denoted $\hat{R}_{U,ML}$, is given by the standard sample covariance matrix in (6.5.160),

$$\boxed{\hat{R}_{U,ML} = \hat{R}} \tag{6.5.162}$$

whereas the persymmetric ML estimate of R, denoted $\hat{R}_{P,ML}$, is given by

$$\boxed{\hat{R}_{P,ML} = \frac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right)} \tag{6.5.163}$$

To prove (6.5.162), we need to show (see (6.5.159)) that

$$\ln|R| + \operatorname{tr}\left(R^{-1}\hat{R}\right) \geq \ln|\hat{R}| + m \quad \text{for any } R > 0 \tag{6.5.164}$$

Let $\hat{C}$ be a square root of $\hat{R}$ (see Definition D12 in Appendix A) and note that

$$\operatorname{tr}\left(R^{-1}\hat{R}\right) = \operatorname{tr}\left(R^{-1}\hat{C}\hat{C}^*\right) = \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) \tag{6.5.165}$$

Using (6.5.165) in (6.5.164), we obtain the series of equivalences

$$\begin{aligned}
(6.5.164) \iff & \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) - \ln\left|R^{-1}\hat{R}\right| \geq m \\
\iff & \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) - \ln\left|\hat{C}^*R^{-1}\hat{C}\right| \geq m \\
\iff & \sum_{k=1}^{m}(\lambda_k - \ln\lambda_k - 1) \geq 0 \qquad (6.5.166)
\end{aligned}$$

where $\{\lambda_k\}$ are the eigenvalues of the matrix $\hat{C}^*R^{-1}\hat{C}$.

Next, with reference to (6.5.166), we show that

$$f(\lambda) \triangleq \lambda - \ln\lambda - 1 \geq 0 \quad \text{for any } \lambda > 0 \tag{6.5.167}$$

To verify (6.5.167), observe that

$$f'(\lambda) = 1 - \frac{1}{\lambda}; \qquad f''(\lambda) = \frac{1}{\lambda^2}$$

Hence, the function $f(\lambda)$ in (6.5.167) has a unique minimum at $\lambda = 1$, and $f(1) = 0$; this proves (6.5.167). With this observation, the proof of (6.5.166), and therefore of (6.5.162), is complete.

The proof of (6.5.163) is even simpler. In view of (6.5.161), we have that

$$\begin{aligned}
\operatorname{tr}\left(R^{-1}\hat{R}\right) &= \operatorname{tr}\left[\left(JR^TJ\right)^{-1}\hat{R}\right] = \operatorname{tr}\left(R^{-T}J\hat{R}J\right) \\
&= \operatorname{tr}\left(R^{-1}J\hat{R}^TJ\right) \qquad (6.5.168)
\end{aligned}$$

Hence, the function to be minimized with respect to R (under the constraint (6.5.161)) can be written as

$$\ln|R| + \operatorname{tr}\left[R^{-1}\cdot\tfrac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right)\right] \tag{6.5.169}$$

As was shown earlier in this complement, the unstructured minimizer of (6.5.169) is given by

$$R = \frac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right) \tag{6.5.170}$$

Because (6.5.170) satisfies the persymmetry constraint, by construction, it also gives the constrained minimizer of the negative log-likelihood function; hence, the proof of (6.5.163) is concluded as well.

The reader interested in more details on the topic of this complement, including a comparison of the statistical estimation errors associated with $\hat{R}_{U,ML}$ and $\hat{R}_{P,ML}$, can consult [Jansson and Stoica 1999].

6.6 EXERCISES

Exercise 6.1: Source Localization by Using a Sensor in Motion

This exercise illustrates how the directions of arrival of planar waves can be found by using a single moving sensor. Conceptually, this problem is related to that of DOA estimation by sensor-array methods. Indeed, we can think of a sensor in motion as creating a *synthetic aperture* similar to the one corresponding to a physical array of spatially distributed sensors.

Assume that the sensor has a linear motion with constant speed equal to v. Also, assume that the sources are far-field point emitters at fixed locations in the same plane as the sensor. Let θ_k denote the kth DOA parameter (defined as the angle between the direction of wave propagation and the normal to the sensor trajectory). Finally, assume that the sources emit sinusoidal signals $\{\alpha_k e^{i\omega t}\}_{k=1}^{n}$ with the same (center) frequency ω. (These signals could be reflections of a probing sinusoidal signal from different point scatterers of a target, in which case it is not restrictive to assume that they all have the same frequency.)

Show that, under the previous assumptions and after elimination of the high-frequency component corresponding to the frequency ω, the sensor output signal can be written as

$$s(t) = \sum_{k=1}^{n} \alpha_k e^{i\omega_k^D t} + e(t) \tag{6.6.1}$$

where $e(t)$ is measurement noise and ω_k^D is the kth *Doppler frequency*, defined by

$$\omega_k^D = -\frac{v \cdot \omega}{c} \sin\theta_k$$

with c denoting the velocity of signal propagation. Conclude from (6.6.1) that the DOA-estimation problem associated with the scenario under consideration can be solved by using the estimation methods discussed in this chapter and in Chapter 4 (provided that the sensor speed v can be accurately determined).

Exercise 6.2: Beamforming Resolution for Uniform Linear Arrays

Consider a ULA comprising m sensors, with interelement spacing equal to d. Let λ denote the wavelength of the signals impinging on the array. According to the discussion in Chapter 2, the *spatial-frequency resolution* of the beamforming used with this ULA is given by

$$\Delta\omega_s = \frac{2\pi}{m} \qquad \Longleftrightarrow \qquad \Delta f_s = \frac{1}{m} \tag{6.6.2}$$

Make use of the previous observation to show that the *DOA resolution* of beamforming for signals coming from *broadside* is

$$\boxed{\Delta\theta \simeq \sin^{-1}(1/L)} \tag{6.6.3}$$

where L is the array's length measured in wavelengths:

$$\boxed{L = \frac{(m-1)d}{\lambda}} \tag{6.6.4}$$

Explain how (6.6.3) approximately reduces to (6.3.20), for sufficiently large L.

Next, show that, for signals impinging from an *arbitrary direction angle* θ, the *DOA resolution* of beamforming is approximately

$$\boxed{\Delta\theta \simeq \frac{1}{L|\cos\theta|}} \tag{6.6.5}$$

Hence, for signals coming from nearly end-fire directions, the DOA resolution is much worse than what is suggested in (6.3.20).

Exercise 6.3: Beamforming Resolution for Arbitrary Arrays
The *beampattern*

$$W(\theta) = |a^*(\theta)a(\theta_0)|^2, \qquad \text{(some } \theta_0\text{)}$$

has the same shape as a spectral window: It has a peak at $\theta = \theta_0$, is symmetric about that point, and the peak is narrow (for large enough values of m). Consequently, the *beamwidth* of the array with direction vector $a(\theta)$ can approximately be derived by using the window bandwidth formula proven in Exercise 2.15:

$$\boxed{\Delta\theta \simeq 2\sqrt{|W(\theta_0)/W''(\theta_0)|}} \tag{6.6.6}$$

Now, the array's beamwidth and the resolution of beamforming are closely related. To see this, consider the case where the array output covariance matrix is given by (6.4.3). Let $n = 2$, and assume that $P = I$ (for simplicity of explanation). The average beamforming spectral function is then given by

$$a^*(\theta)Ra(\theta) = |a^*(\theta)a(\theta_1)|^2 + |a^*(\theta)a(\theta_2)|^2 + m\sigma^2$$

which clearly shows that the sources with DOAs θ_1 and θ_2 are resolvable by beamforming if and only if $|\theta_1 - \theta_2|$ is larger than the array's beamwidth. Consequently, we can approximately determine the beamforming resolution by using (6.6.6). Specialize equation (6.6.6) to a ULA, and compare with the results obtained in Exercise 6.2.

Exercise 6.4: Beamforming Resolution for L-Shaped Arrays
Consider an m-element array, with m odd, shaped as an "L" with element spacing d. Thus, the array elements are located at points $(0,0)$, $(0,d),\ldots,(0,d(m-1)/2)$ and $(d,0),\ldots,(d(m-$

1)/2, 0). Using the results in Exercise 6.3, find the DOA resolution of beamforming for signals coming from an angle θ. What are the minimum and maximum resolution, and for what angles are these extremal resolutions realized? Compare your results with the m-element ULA case in Exercise 6.2.

Exercise 6.5: Relationship between Beamwidth and Array-Element Locations
Consider an m-element planar array with elements located at $r_k = [x_k, y_k]^T$ for $k = 1, \ldots, m$. Assume that the array is centered at the origin, so $\sum_{k=1}^{m} r_k = 0$. Use equation (6.6.6) to show that the array beamwidth at direction θ_0 is given by

$$\Delta\theta \simeq \sqrt{2}\frac{\lambda}{2\pi}\frac{1}{D(\theta_0)} \tag{6.6.7}$$

where $D(\theta_0)$ is the root-mean-square distance of the array elements to the origin in the direction orthogonal to θ_0 (see Figure 6.8):

$$D(\theta_0) = \sqrt{\frac{1}{m}\sum_{k=1}^{m} d_k^2(\theta_0)}, \qquad d_k(\theta_0) = x_k \sin\theta_0 - y_k \cos\theta_0$$

As in Exercise 2.15, the beamwidth approximation in equation (6.6.7) slightly underestimates the true beamwidth; a better approximation is given by

$$\boxed{\Delta\theta \simeq 1.15\sqrt{2}\frac{\lambda}{2\pi}\frac{1}{D(\theta_0)}} \tag{6.6.8}$$

Exercise 6.6: Isotropic Arrays
An array whose beamwidth is the same for all directions is said to be *isotropic*. Consider an m-element planar array with elements located at $r_k = [x_k, y_k]^T$ for $k = 1, \ldots, m$ and centered at

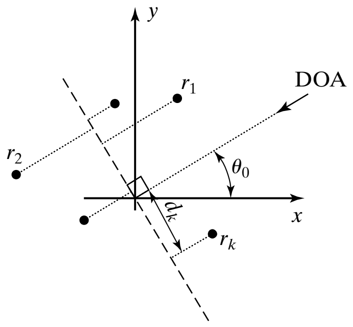


Figure 6.8 Array-element projected distances from the origin for DOA angle θ_0 (see Exercise 6.5).

the origin ($\sum_{k=1}^{m} r_k = 0$), as in Exercise 6.5. Show that the array beamwidth (as given by (6.6.7)) is the same for all DOAs if and only if

$$R^T R = cI_2 \tag{6.6.9}$$

where

$$R = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix}$$

and where c is a positive constant. (See [Baysal and Moses 2003] for additional details and properties of isotropic arrays.)

Exercise 6.7: Grating Lobes

The results of Exercise 6.2 might suggest that an m-element ULA can have very high resolution simply by using a large array element spacing d. However, there is an ambiguity associated with choosing $d > \lambda/2$; this drawback is sometimes referred to as the problem of *grating lobes*. Identify this drawback, and discuss what ambiguities exist as a function of d. (Refer to the discussion on ULAs in Section 6.2.2.)

One potential remedy to this drawback is to use two ULAs: one with m_1 elements and element spacing $d_1 = \lambda/2$, and another with m_2 elements and element spacing d_2. Discuss how to choose m_1, m_2, and d_2 to both avoid ambiguities and increase resolution over a conventional ULA with element spacing $d = \lambda/2$ and $m_1 + m_2$ elements. Consider as an example using a 10-element ULA with $d_2 = 3\lambda/2$ for the second ULA; find m_1 to resolve ambiguities in this array. Finally, discuss any potential drawbacks of the two-array approach.

Exercise 6.8: Beamspace Processing

Consider an array comprising many sensors ($m \gg 1$). Such an array should be able to resolve sources that are quite closely spaced (*cf.* (6.3.20) and the discussion in Exercise 6.3). There is, however, a price to be paid for the high-resolution performance achieved by using many sensors: the computational burden associated with the *elementspace processing* (ESP) (i.e., the direct processing of the output of all sensors) may be prohibitively high, and the involved circuitry (A–D converters, etc.) could be quite expensive.

Let B^* be an $\bar{m} \times m$ matrix with $\bar{m} < m$, and consider the transformed output vector $B^* y(t)$. The latter vector satisfies the following equation (*cf.* (6.2.21)):

$$B^* y(t) = B^* A s(t) + B^* e(t) \tag{6.6.10}$$

The transformation matrix B^* can be interpreted as a *beamformer* or *spatial filter* acting on $y(t)$. Estimation of the DOAs of the signals impinging on the array using $B^* y(t)$ is called *beamspace processing* (BSP). Since $\bar{m} < m$, BSP should have a lower computational burden than ESP. The critical question is then how to choose the beamformer B so as not to significantly degrade the performance achievable by ESP.

Assume that a certain DOA sector is known to contain the source(s) of interest (whose DOAs are designated by the generic variable θ_0). Using this information, design a matrix B^* that passes the signals from direction θ_0 approximately undistorted. Choose B in such a way that the noise in beamspace, $B^*e(t)$, is still spatially white. For a given sector size, discuss the tradeoff between the computational burden associated with BSP and the distorting effect of the beamformer on the desired signals. Finally, use the results of Exercise 6.3 to show that the resolution of beamforming in elementspace and beamspace are nearly the same, under the previous conditions.

Exercise 6.9: Beamspace Processing (cont'd)

In this exercise, for simplicity, we consider the beamspace processing (BSP) equation (6.6.10) for the case of a single source ($n = 1$):

$$B^*y(t) = B^*a(\theta)s(t) + B^*e(t) \tag{6.6.11}$$

The elementspace processing (ESP) counterpart of (6.6.11) (*cf.* (6.2.19)) is

$$y(t) = a(\theta)s(t) + e(t) \tag{6.6.12}$$

Assume that $\|a(\theta)\|^2 = m$ (see (6.3.11)) and that the $\bar{m} \times m$ matrix B^* is unitary (i.e., $B^*B = I$). Furthermore, assume that

$$a(\theta) \in \mathcal{R}(B) \tag{6.6.13}$$

To satisfy (6.6.13), we need knowledge about a DOA sector that contains θ, which is usually assumed to be available in BSP applications; note that the narrower this sector, the smaller the value we can choose for $\bar{m}$. As $\bar{m}$ decreases, the implementation advantages of BSP compared with ESP become more significant. However, the DOA estimation performance achievable by BSP might be expected to decrease as $\bar{m}$ decreases. As indicated in Exercise 6.8, this is not necessarily the case. In the present exercise, we lend further support to the fact that the estimation performances of ESP and BSP can be quite similar to one another, provided that condition (6.6.13) is satisfied. To be specific, define the array SNR for (6.6.12) as

$$\frac{E\left\{\|a(\theta)s(t)\|^2\right\}}{E\|e(t)\|^2} = \frac{mP}{m\sigma^2} = \frac{P}{\sigma^2} \tag{6.6.14}$$

where P denotes the power of $s(t)$. Show that the "array SNR" for the BSP equation, (6.6.11), is $m/\bar{m}$ times that in (6.6.14). Conclude that this increase in the array SNR associated with BSP might counterbalance the presumably negative impact on DOA performance due to the decrease from m to $\bar{m}$ in the number of observed output signals.

Exercise 6.10: Beamforming and MUSIC under the Same Umbrella

Define the scalars

$$Y_t^*(\theta) = a^*(\theta)y(t), \qquad t = 1, \ldots, N.$$

By using previous notation, we can write the beamforming spatial spectrum in (6.3.18) as

$$Y^*(\theta)WY(\theta) \tag{6.6.15}$$

where

$$W = (1/N)I \qquad \text{(for beamforming)}$$

and

$$Y(\theta) = [Y_1(\theta) \dots Y_N(\theta)]^T$$

Show that the MUSIC spatial pseudospectrum

$$a^*(\theta)\hat{S}\hat{S}^*a(\theta) \tag{6.6.16}$$

(see Sections 4.5 and 6.4.3) can also be put in the form (6.6.15), for a certain "weighting matrix" W. The columns of the matrix $\hat{S}$ in (6.6.16) are the n principal eigenvectors of the sample covariance matrix $\hat{R}$ in (6.3.17).

Exercise 6.11: Subspace Fitting Interpretation of MUSIC

In words, the result (4.5.9) (on which MUSIC for both frequency and DOA estimation is based) says that the direction vectors $\{a(\theta_k)\}$ belong to the subspace spanned by the columns of S. Therefore, we can think of estimating the DOAs by choosing θ (a generic DOA variable) so that the distance between $a(\theta)$ and the closest vector in the span of $\hat{S}$ is minimized—that is,

$$\min_{\beta,\theta} \|a(\theta) - \hat{S}\beta\|^2 \tag{6.6.17}$$

where $\|\cdot\|$ denotes the Euclidean vector norm. Note that the dummy vector variable β in (6.6.17) is defined in such a way that $\hat{S}\beta$ is closest to $a(\theta)$ in Euclidean norm.

Show that the DOA estimation method derived from the subspace-fitting criterion (6.6.17) is the same as MUSIC.

Exercise 6.12: Subspace Fitting Interpretation of MUSIC (cont'd)

The result (4.5.9) can also be invoked to arrive at the following subspace fitting criterion:

$$\min_{B,\theta} \|A(\theta) - \hat{S}B\|_F^2 \tag{6.6.18}$$

where $\|\cdot\|_F$ stands for the Frobenius matrix norm and θ is now the vector of all DOA parameters. This criterion seems to be a more general version of equation (6.6.17) in Exercise 6.11. Show that the minimization of the multidimensional subspace fitting criterion in (6.6.18), with respect to the DOA vector θ, still leads to the one-dimensional MUSIC method. **Hint:** It will be useful to refer to the type of result proven in equations (4.3.12)–(4.3.16) in Section 4.3.

Exercise 6.13: Subspace Fitting Interpretation of MUSIC (cont'd)
The subspace fitting interpretations of the previous two exercises provide some insights into the properties of the MUSIC estimator. Assume, for instance, that two or more source signals are coherent. Make use of the subspace fitting interpretation in Exercise 6.12 to show that MUSIC cannot be expected to yield meaningful results in such a case. Follow the line of your argument, explaining why MUSIC fails in the case of coherent signals, to suggest a subspace fitting criterion that works in such a case. Discuss the computational complexity of the method based on the latter criterion.

Exercise 6.14: Modified MUSIC for Coherent Signals
Consider an m-element ULA. Assume that n signals impinge on the array at angles $\{\theta_k\}_{k=1}^n$ and, also, that some signals are coherent (so that the signal covariance matrix P is singular). Derive a modified MUSIC DOA estimator for this case, analogous to the modified MUSIC frequency estimator in Section 4.5, and show that this method is capable of estimating the n DOAs even in the coherent-signal case.

COMPUTER EXERCISES

Tools for Array Signal Processing:
The text website `www.prenhall.com/stoica` contains the following MATLAB functions for use in DOA estimation:

- `Y=uladata(theta,P,N,sig2,m,d)`
  Generates an $m \times N$ data matrix $Y = [y(1), \dots, y(N)]$ for a ULA with `n` sources arriving at angles (in degrees from $-90°$ to $90°$) given by the elements of the $n \times 1$ vector `theta`. The source signals are zero-mean Gaussian with covariance matrix $P = E\{s(t)s^*(t)\}$. The noise component is spatially white Gaussian with covariance $\sigma^2 I$, where $\sigma^2 =$`sig2`. The element spacing is equal to `d` in wavelengths.
- `phi=beamform(Y,L,d)`
  Implements the beamforming spatial spectral estimate in equation (6.3.18) for an m-element ULA with sensor spacing `d` in wavelengths. The $m \times N$ matrix `Y` is as defined above. The parameter `L` controls the DOA sampling, and `phi` is the spatial spectral estimate `phi`$=[\hat{\phi}(\theta_1), \dots, \hat{\phi}(\theta_L)]$, where $\theta_k = -\frac{\pi}{2} + \frac{\pi k}{L}$.
- `phi=capon_sp(Y,L,d)`
  Implements the Capon spatial spectral estimator in equation (6.3.26); the input and output parameters are defined as those in `beamform`.
- `theta=root_music_doa(Y,n,d)`
  Implements the Root MUSIC method in Section 4.5, adapted for spatial spectral estimation using a ULA. The parameters `Y` and `d` are as in `beamform`, and `theta` is the vector containing the `n` DOA estimates $[\hat{\theta}_1, \dots, \hat{\theta}_n]^T$.
- `theta=esprit_doa(Y,n,d)`
  Implements the ESPRIT method for a ULA. The parameters `Y` and `d` are as in `beamform`, and `theta` is the vector containing the `n` DOA estimates $[\hat{\theta}_1, \dots, \hat{\theta}_n]^T$. The two subarrays for ESPRIT are made from the first $m - 1$ and last $m - 1$ elements of the array.

Exercise C6.15: Comparison of Spatial Spectral Estimators

Simulate the following scenario: Two signals with wavelength λ impinge on an array of sensors from DOAs $\theta_1 = 0°$ and a θ_2 that will be varied. The signals are mutually uncorrelated complex Gaussian with unit power, so that $P = E\{s(t)s^*(t)\} = I$. The array is a 10-element ULA with element spacing $d = \lambda/2$. The measurements are corrupted by additive complex Gaussian white noise with unit power. $N = 100$ snapshots are collected.

(a) Let $\theta_2 = 15°$. Compare the results of the beamforming, Capon, Root MUSIC, and ESPRIT methods for this example. The results can be shown by plotting the spatial spectrum estimates from beamforming and Capon for 50 Monte Carlo experiments; for Root MUSIC and ESPRIT, plot vertical lines of equal height located at the DOA estimates from the 50 Monte Carlo experiments. How do the methods compare? Are the properties of the various estimators analogous to the time-series case for two sinusoids in noise?

(b) Repeat for $\theta_2 = 7.5°$.

Exercise C6.16: Performance of Spatial Spectral Estimators for Coherent Source Signals

In this exercise, we will see what happens when the source signals are fully correlated (or coherent). Use the same parameters and estimation methods as in Exercise C6.15 with $\theta_2 = 15°$, but with

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Note that the sources are coherent as $\text{rank}(P) = 1$.

Compare the results of the four methods for this case, again by plotting the spatial spectrum and "DOA line spectrum" estimates (as in Exercise C6.15) for 50 Monte Carlo experiments from each estimator. Which method appears to be the best in this case?

Exercise C6.17: Spatial Spectral Estimators Applied to Measured Data

Apply the four DOA estimators from Exercise C6.15 to the real data in the file `submarine.mat`, which can be found at the text website `www.prenhall.com/stoica`. These data are underwater measurements collected by the Swedish Defense Agency in the Baltic Sea. The 6-element array of hydrophones used in the experiment can be assumed to be a ULA with inter-element spacing equal to 0.9 m. The wavelength of the signal is approximately 5.32 m. Can you find the "submarine(s)?"