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 non-adaptive 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 following weighted LS fitting problem

\[
\min_a \sum_{t=1}^{N} \rho(t) |y(t) - ae^{i\omega t}|^2
\]

(2.8.10)

where \(\omega\) is given and so are the weights \(\rho(t) \geq 0\). It can be readily verified that the minimizer of (2.8.10) is given by

\[
\hat{a} = \frac{\sum_{t=1}^{N} \rho(t)y(t)e^{-i\omega t}}{\sum_{t=1}^{N} \rho(t)}
\]

(2.8.11)

If we let

\[
v(t) = \frac{\rho(t)}{\sum_{t=1}^{N} \rho(t)}
\]

(2.8.12)

then we can rewrite (2.8.11) as a windowed DFT

\[
\hat{a} = \sum_{t=1}^{N} v(t)y(t)e^{-i\omega t}
\]

(2.8.13)
Chapter 2  Nonparametric Methods

The squared magnitude of (2.8.13) appears in the windowed periodogram formula (2.6.24), which of course is not accidental as $|\hat{a}|^2$ should indicate the power in $y(t)$ at frequency $\omega$ (cf. (2.8.10)).

The usefulness of the LS-based derivation of (2.6.24) above lies in the fact that it reveals two constraints which must be satisfied by a temporal window:

$$v(t) \geq 0$$  \hspace{1cm} (2.8.14)

which follows from $\rho(t) \geq 0$, and

$$\sum_{i=1}^{N} v(t) = 1$$  \hspace{1cm} (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 $\omega$ 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 $f_v(t)$ we use the notation $\hat{\psi_v}(\omega)$:

$$\hat{\psi_v}(\omega) = N \left| \sum_{i=1}^{N} v(t)y(t)e^{i\omega t} \right|^2$$  \hspace{1cm} (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 $\sum_{i=1}^{N} v(t) = N$.

In the apodization approach the window is selected such that

$$\hat{\psi_v}(\omega) = \text{minimum}$$  \hspace{1cm} (2.8.17)

for each $\omega$ 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 $C$ denote the class of windows over which we perform the minimization in (2.8.17). Each window in $C$ must satisfy the constraints (2.8.14) and (2.8.15). Usually, $C$ 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 $v(t) \in C$. 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, $\hat{\psi_v}(\omega)$
depends quadratically on more than one parameter, then in general the minimum
value will be zero, $\hat{v}(!t) = 0$ for all $\omega$, which is not acceptable. We postpone a
more detailed discussion on the parameterization of $C$ 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 rect-
angular 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 correspond-
ing periodogram) for less leakage. By using the apodization principle in (2.8.17)
to choose between $\hat{v}_1(\omega)$ and $\hat{v}_2(\omega)$, at each frequency $\omega$, we obtain the spectral
estimate shown in Figure 2.5, which inherits the high resolution of $\hat{v}_1(\omega)$ and the
low leakage of $\hat{v}_2(\omega)$.

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

$$\sum_{t=1}^{N} h_t e^{i\omega t} = 1 \quad (2.8.18)$$

and hence the apodization design problem is to minimize

$$\left| \sum_{t=1}^{N} h_t y(t) \right|^2 \quad (2.8.19)$$

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.
subject to (2.8.18) as well as (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 \( N \), and consequently (2.8.19) is the “power” of the filter output and (2.8.18) is the (complex) gain of the filter at frequency \( \omega \). Therefore, making use of \( \{ h(t) \} \), we can describe the apodization principle in words as follows: find the (parameterized) FIR filter \( \{ h(t) \} \) which passes without distortion the sinusoid with frequency \( \omega \) (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 \( \omega \). This interpretation can clearly 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 apparently noted for the first time in [Lee and Munson Jr. 1995]).

As already stated above, an important aspect that remains to be discussed is the parameterization of \( \{ v(t) \} \). For the apodization principle to make sense, the class \( C \) of windows must be chosen carefully. In particular, as explained above, 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 \( C \) is also important from a computational standpoint. Indeed, the task of solving (2.8.17), for each \( \omega \), and then computing the corresponding \( \hat{\phi}_v(\omega) \) may be computationally demanding unless \( C \) is carefully chosen.

In the following we will consider the class of temporal windows used in [Stankwitz, Dallaire, and Fienup 1994]:

\[
v(t) = \frac{1}{N} \left[ \alpha - \beta \cos \left( \frac{2\pi}{N} t \right) \right], \quad t = 1, \ldots, N \quad (2.8.20)
\]

It can be readily checked that (2.8.20) satisfies the constraints (2.8.14) and (2.8.15) if and only if

\[
\alpha = 1 \quad \text{and} \quad |\beta| \leq 1 \quad (2.8.21)
\]

In addition we require that

\[
\beta \geq 0 \quad (2.8.22)
\]

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, and 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 \([-N, N]\) and 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:

\[ v(t) = \frac{1}{N} \left[ 1 - \beta \cos \left( \frac{2\pi}{N} t \right) \right] \]
\[ = \frac{1}{N} \left[ 1 - \frac{\beta}{2} \left( e^{i \frac{2\pi}{N} t} + e^{-i \frac{2\pi}{N} t} \right) \right], \quad \beta \in [0, 1] \quad (2.8.23) \]

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

\[ Y(k) = \sum_{t=1}^{N} y(t) e^{-i \frac{2\pi}{N} k t}, \quad k = 1, \ldots, N \quad (2.8.24) \]

(see Section 2.3). Then the windowed periodogram corresponding to (2.8.23) can be conveniently computed as follows:

\[ \hat{\phi}_w(k) = \frac{1}{N} \left| Y(k) - \frac{\beta}{2} \left[ Y(k-1) + Y(k+1) \right] \right|^2, \quad k = 2, \ldots, N-1 \quad (2.8.25) \]

Furthermore, in (2.8.25) \( \beta \) is the solution to the following apodization design problem:

\[ \min_{\beta \in [0, 1]} \left| Y(k) - \frac{\beta}{2} \left[ Y(k-1) + Y(k+1) \right] \right|^2 \quad (2.8.26) \]

The unconstrained minimizer of the above function is given by:

\[ \beta_0 = \text{Re} \left( \frac{2Y(k)}{Y(k-1) + Y(k+1)} \right) \quad (2.8.27) \]

Because the function in (2.8.26) is quadratic in \( \beta \), it follows that the constrained minimizer of (2.8.26) is given by

\[ \beta = \begin{cases} 
0, & \text{if } \beta_0 < 0 \\
\beta_0, & \text{if } 0 \leq \beta_0 \leq 1 \\
1, & \text{if } \beta_0 > 1
\end{cases} \quad (2.8.28) \]

**Remark:** It is interesting to note from (2.8.28) that a change of the value of \( \alpha \) 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 \( \alpha \), for instance to \( \alpha = 0.75 \), then the interval for \( \beta \) 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 simply obtained by using \( \beta \) given by (2.8.28) in (2.8.25). Hence, despite the fact that the apodization window is both frequency and data dependent (via \( \beta \) in (2.8.27), (2.8.28)) the implementation...
of the corresponding spectral estimate is only marginally more computationally demanding than 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 when studying 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

\[
\hat{\phi}(\omega) = \frac{1}{N} Z(\omega)Z^*(\omega)
\]

(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

\[
\hat{\phi}(\omega) = \left[ \begin{array}{cc} \hat{\phi}_{yy}(\omega) & \hat{\phi}_{yu}(\omega) \\ \hat{\phi}_{yu}^*(\omega) & \hat{\phi}_{uu}(\omega) \end{array} \right]
\]

(2.8.30)

As indicated by the notation previously used, estimates of \( \hat{\phi}_{yy}(\omega) \), \( \hat{\phi}_{uu}(\omega) \) and of the cross–spectrum \( \hat{\phi}_{yu}(\omega) \) may be obtained from the corresponding elements of \( \hat{\phi}(\omega) \).

We first show that the estimate of the coherency spectrum obtained from (2.8.30) is always such that

\[
|\hat{C}_{yu}(\omega)| = 1 \quad \text{for all } \omega
\]

(2.8.31)

and hence it is useless. To see this, note that since the rank of the \( 2 \times 2 \) matrix in (2.8.30) is equal to one (see Result R22 in Appendix A), we must have

\[
\hat{\phi}_{uu}(\omega)\hat{\phi}_{yu}(\omega) = |\hat{\phi}_{yu}(\omega)|^2
\]

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. We also conclude from this fact that the unwindowed periodogram is a poor estimate of the PSD.

Consider next a windowed Blackman–Tukey periodogram estimate of the cross–spectrum:

\[
\hat{\phi}_{yu}(\omega) = \sum_{k=-M}^{M} w(k)\hat{r}_{yu}(k)e^{-i\omega k}
\]

(2.8.32)
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 is shown above 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

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

(3.9.23)

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

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

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

(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 $\omega$) 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 $\omega$) 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 determined 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...]}
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 \( m \):

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

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

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

The order \( m \) in (3.9.25) may 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, e.g., [WILSON 1969; VOSTRY 1975; VOSTRY 1976]), and in this way derive MA parameter estimates \( \{b_k\} \) satisfying

\[
\phi(\omega) = \sigma^2 |B(\omega)|^2
\] (3.9.27)

(see (3.6.1)). This step of obtaining \( \{b_k\} \) and \( \sigma^2 \) from \( \phi(\omega) \) can be computed in \( O(m^2) \) flops. The problem that remains is to solve (3.9.26) for \( \phi(\omega) \) in a similarly efficient computational way.

As

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

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:

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

Consequently, the approximation problem (3.9.26) is equivalent to:

\[
\min_{\{\hat{r}(k)\}} \|\hat{r} - r\|_W^2, \text{ subject to } (3.9.25)
\] (3.9.28)
where \( \|x\|_W^2 = x^* W x \) and

\[
\hat{r} = \begin{bmatrix} \hat{r}(0) & \cdots & \hat{r}(m) \end{bmatrix}^T \\
\hat{r} = \begin{bmatrix} r(0) & \cdots & r(m) \end{bmatrix}^T \\
W = \begin{bmatrix} 1 & 0 \\
0 & 2 \\
. & . \\
0 & 2 \end{bmatrix}
\]

In the following 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). Note that a possible way of tackling (3.9.28) would consist of writing the covariances \( \{r(k)\} \) as functions of the MA parameters (see (3.3.3)), which would guarantee that they satisfy (3.9.25), and then minimize the function in (3.9.28) with respect to the MA parameters. However, the so-obtained minimization problem would be, similarly to (3.9.23), nonlinear in the MA parameters (more precisely, the criterion in (3.9.28) is quartic in \( \{b_k\} \)), which is exactly the type of problem we tried to avoid in the first place.

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

\[
J_k = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 \\
. & . & . & . & . \\
0 & \cdots & 1 \\
. & . & . & . & . \\
0 & \cdots & \cdots & \cdots \end{bmatrix}, \quad (m+1) \times (m+1)
\]

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

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

To prove this result, let

\[
a(\omega) = \begin{bmatrix} 1 & e^{i\omega} & \cdots & e^{im\omega} \end{bmatrix}^T
\]
and observe that

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

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

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

which implies that \(\{r(k)\}\) indeed is an MA\((m)\) covariance sequence. To show that any MA\((m)\) covariance sequence can be parameterized as in (3.9.29), we make use of (3.3.3) to write (for \(k = 0, \ldots, m\))

\[
r(k) = \sigma^2 \sum_{j=k}^{m} b_j b^*_j = \sigma^2 \begin{bmatrix} b_0^* & \cdots & b_m^* \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} = \text{tr} \left\{ J_k \cdot \sigma^2 \begin{bmatrix} b_0 \\ \vdots \\ b_m \end{bmatrix} [ b_0^* & \cdots & b_m^* ] \right\} \quad (3.9.30)
\]

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

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

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

\[
\mu = C(\hat{\rho} - r)
\]

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

\[
\alpha = [Q_{1,1} \ Q_{1,2} \ \cdots \ Q_{1,m+1} ; \ Q_{2,2} \ \cdots \ Q_{2,m+1} ; \ \cdots ; \ Q_{m+1,m+1}]^T
\]
Section 3.10 Exercises

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

\[
\begin{array}{ll}
\min_{\rho, \mu, \alpha} & \rho \\
\text{subject to:} & \|\mu\| \leq \rho \\
& Q \succeq 0 \\
& \begin{bmatrix}
\text{tr}[Q] \\
\text{tr} \left[ \frac{1}{2} (J_1 + J_1^T)Q \right] \\
\vdots \\
\text{tr} \left[ \frac{1}{2} (J_m + J_m^T)Q \right]
\end{bmatrix} + C^{-1}\mu = \hat{r}
\end{array}
\]

(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):

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

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 efficiently solved 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 \( r(k) = \text{tr}(J_kQ) \) for \( k = 0, \ldots, m \). Numerical results obtained with the MA parameter estimation algorithm outlined above 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 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 \( A(z) = 1 + a_1z^{-1} + \cdots + a_nz^{-n} \) 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 \( B(0) = A(0) \) (where \( B(\omega) \triangleq B(z) \bigr|_{z = e^{i\omega}} \)), then the phase lag of \( B(\omega) \), given by
Chapter 4 Parametric Methods for Line Spectra

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 \( H_1 \) above; 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 \( \alpha(t) \in \mathbb{R} \) is an arbitrary unknown envelope modulating the sinusoidal signal. The NLS estimates of \( \alpha(t) \), \( \omega \), and \( \varphi \) are obtained by minimizing the following 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 the above 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:

\[
f = \sum_{t=1}^{N} \left( |y(t)|^2 + \left[ \alpha(t) - \text{Re} \left( e^{-i(\omega t + \varphi)}y(t) \right) \right]^2 - \left[ \text{Re} \left( e^{-i(\omega t + \varphi)}y(t) \right) \right]^2 \right)
\]

(4.9.19)

The minimization of (4.9.19) with respect to \( \alpha(t) \) is immediate:

\[
\hat{\alpha}(t) = \text{Re} \left( e^{-i(\hat{\omega} t + \hat{\varphi})}y(t) \right) \tag{4.9.20}
\]

where 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 \( \varphi \) and \( \omega \) are obtained by maximizing the function

\[
g = 2 \sum_{t=1}^{N} \left[ \text{Re} \left( e^{-i(\omega t + \varphi)}y(t) \right) \right]^2
\]

where the factor 2 has been introduced for the sake of convenience. For any complex number \( c \) we have

\[
|\text{Re}(c)|^2 = \frac{1}{4} (c + c^*)^2 = \frac{1}{2} \left[ |c|^2 + \text{Re} \left( c^2 \right) \right]
\]
Section 4.9 Complements

It follows that

\[ g = \sum_{t=1}^{N} \left( |y(t)|^2 + \text{Re} \left[ e^{-2i(\omega t + \varphi)} y^2(t) \right] \right) \]

\[ = \text{constant} + \sum_{t=1}^{N} y^2(t)e^{-i2\omega t} \cdot \cos \left[ \text{arg} \left( \sum_{t=1}^{N} y^2(t)e^{-i2\omega t} \right) - 2\varphi \right] \]  

(4.9.21)

Clearly the maximizing \( \varphi \) is given by

\[ \hat{\varphi} = \frac{1}{2} \text{arg} \left( \sum_{t=1}^{N} y^2(t)e^{-i2\omega t} \right) \]

with the NLS estimate of \( \omega \) given by

\[ \hat{\omega} = \text{arg max}_{\omega} \left| \sum_{t=1}^{N} y^2(t)e^{-i2\omega t} \right| \]  

(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 \( \pi \). The restriction of \( \omega \) to \([0, \pi]\) is not a peculiar feature of the NLS approach, but 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

\[ y(t) = \hat{\alpha}(t)e^{i(\omega t + \varphi)} + e(t), \quad t = 1, \ldots, N \]

where \( \hat{\alpha}(t) = (-1)^t\alpha(t) \) is another valid (i.e., real-valued) envelope. This simple calculation confirms the fact that \( \omega \) 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 above estimate of \( \omega \) should be contrasted with the NLS estimate of \( \omega \) in the constant-amplitude case (see (4.3.11), (4.3.17)):

\[ \hat{\omega} = \text{arg max}_{\omega} \left| \sum_{t=1}^{N} y(t)e^{-i\omega t} \right| \quad (\text{for } \alpha(t) = \text{constant}) \]  

(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 may wonder if 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. Hence the DFT of \( \{\alpha(t)\} \), denoted \( A(\hat{\omega}) \), takes on small values (theoretically zero) at \( \hat{\omega} = 0 \). As the DFT of \( \alpha(t)e^{i\omega t} \) is \( A(\hat{\omega} - \omega) \), it follows that the modulus of this DFT has a valley instead of a peak at
\( \bar{\omega} = \omega \), and hence the standard periodogram (see (4.9.23)) should not be used to determine \( \omega \). On the other hand, \( \alpha^2(t) \) always has a nonzero average value (or DC component), and 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 Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximization (EM) Algorithm

As explained in Section 4.3, minimizing the NLS criterion with respect to the unknown frequencies is a rather difficult task owing to the existence of possibly many local minima and the sharpness of the global minimum. In this complement 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 \( \theta \) 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 \( \theta^i \) denote the value taken by \( \theta \) at the \( i \)th 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:

\[
    f(\theta^{i+1}) \leq f(\theta^i) \quad \text{for } i = 0, 1, 2, \ldots
\]

(4.9.24)

Hereafter \( \theta^0 \) denotes the initial value (or estimate) of \( \theta \) used by the minimization algorithm in question. Clearly (4.9.24) is an appealing property which in effect is the main reason for the interest in the algorithms discussed here. However, we should note that usually (4.9.24) can only guarantee the convergence to a local minimum of \( f(\theta) \). The goodness of the initial estimate \( \theta^0 \) will often determine whether the algorithm will converge to the global minimum. In fact, for some of the algorithms discussed below 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, e.g., [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 \( \theta \) 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:

\[
\begin{align*}
    y^0 &= \text{given} \\
    \text{For } i = 1, 2, \ldots \text{ compute:} \\
    x^i &= \arg \min_x f(x, y^{i-1}) \\
    y^i &= \arg \min_y f(x^i, y)
\end{align*}
\]  

(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 \( x \), and hence the name of “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):

\[
f(x^i, y^i) \leq f(x^i, y^{i-1}) \leq f(x^{i-1}, y^{i-1})
\]

where the first inequality follows from the definition of \( y^i \) and the second from the definition of \( x^i \).

The partitioning of \( \theta \) 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 \( \theta \) in 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 \( \theta \) 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 re-determining \( z \), and so forth.

With reference to the NLS problem in Section 4.3, we can apply the above ideas to the following natural partitioning of the parameter vector:

\[
\begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
\vdots \\
\gamma_n
\end{bmatrix}, \quad \gamma_k = \begin{bmatrix}
\omega_k \\
\varphi_k \\
\alpha_k
\end{bmatrix}
\]  

(4.9.26)

The main virtue of this partitioning of \( \theta \) is that the problem of minimizing the NLS criterion with respect to \( \gamma_k \), 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 simply initialized with \( \gamma_2 = \cdots = \gamma_n = 0 \), in which case \( \gamma_1 \) minimizing the NLS criterion is obtained from the highest peak of the periodogram (which should give a reasonably accurate estimate of \( \gamma_1 \)), and so on.

An elaborated cyclic algorithm, called RELAX, for the minimization of the NLS criterion based on the above ideas (see (4.9.26)), was proposed in [Li AND...
STOICA 1996b]. Note that cyclic minimizers are sometimes called relaxation algorithms, which provides 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, e.g., [HEISER 1995] and the references therein). Assume that, at the $i$th iteration, we can find a function $g_i(\theta)$ (the subindex $i$ indicates the dependence of this function on $\theta^i$) which possesses the following three properties:

$$g_i(\theta^i) = f(\theta^i)$$  \hspace{1cm} (4.9.27)
$$g_i(\theta) \geq f(\theta)$$  \hspace{1cm} (4.9.28)

and

the minimization of $g_i(\theta)$ with respect to $\theta$ is “easy” (or, in any case, easier than the minimization of $f(\theta)$).  \hspace{1cm} (4.9.29)

Owing to (4.9.28), $g_i(\theta)$ is called a majorizing function for $f(\theta)$ at the $i$th iteration. In the majorization technique, the parameter vector at iteration $(i+1)$ is obtained from the minimization of $g_i(\theta)$:

$$\theta^{i+1} = \arg \min_{\theta} g_i(\theta)$$  \hspace{1cm} (4.9.30)

The key property (4.9.24) is satisfied for (4.9.30), since

$$f(\theta^i) = g_i(\theta^i) \geq g_i(\theta^{i+1}) \geq f(\theta^{i+1})$$  \hspace{1cm} (4.9.31)

The first inequality in (4.9.31) follows from the definition of $\theta^{i+1}$ in (4.9.30), and the second inequality from (4.9.28).

Note that any parameter vector $\theta^{i+1}$ which gives a smaller value of $g_i(\theta)$ than $g_i(\theta^i)$ 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 determining $\theta^{i+1}$, for example, by performing a few iterations with a gradient-based algorithm initialized at $\theta^i$ and using a line search (to guarantee that $g_i(\theta^{i+1}) \leq g_i(\theta^i)$). We should note that a similar observation could be made on the cyclic minimizer in (4.9.25) when the minimization of either $f(x, y^{-1})$ 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 may 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 \( \theta \) that makes the minimization in, e.g., (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 may at first seem. Below 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)),

\[
f(\theta) = \sum_{t=1}^{N} [y(t) - \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \varphi_k)}]^2
\]  

where \( \theta \) 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 \( \sigma^2 \). 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 \( \theta \), is

\[
p(y, \theta) = \frac{1}{(\pi \sigma^2)^N} e^{-f(\theta) / \sigma^2}
\]

where \( f(\theta) \) is as defined in (4.9.32) above. The method of maximum likelihood (ML) obtains an estimate of \( \theta \) by maximizing (4.9.33) (see (B.1.7) in Appendix B) or, equivalently, by minimizing the so-called negative log-likelihood function:

\[
-\ln p(y, \theta) = \text{constant} + N \ln \sigma^2 + \frac{f(\theta)}{\sigma^2}
\]

Minimizing (4.9.34) with respect to \( \theta \) 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 \( \theta \) is the negative log-likelihood:

\[
f(\theta) = -\ln p(y, \theta)
\]

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 below concerns the expectation with respect to the distribution of a certain random vector, let us say \( z \), which we will denote by \( E_z \{ \cdot \} \). When the distribution concerned is conditioned on another random vector, let us 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 \( \theta \), then we write \( E_{z|y,\theta} \{ \cdot \} \).

The main result which 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:

\[
E \{ h(x) \} \leq h \left( E \{ x \} \right)
\]  

The proof of (4.9.36) is simple. Let \( d(x) \) denote the plane tangent to \( h(x) \) at the point \( E \{ x \} \). Then

\[
E \{ h(x) \} \leq E \{ d(x) \} = d(E \{ x \}) = h(E \{ x \})
\]

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,

\[
\sum_{k=1}^{M} p_k x_k \geq \prod_{k=1}^{M} x_k^{p_k} \quad \text{(for} \ x_k > 0 \ \text{and} \ \sum_{k=1}^{M} p_k = 1) \quad (4.9.38)
\]

For \( p_k = 1/M \), (4.9.38) reduces to the well-known inequality between the arithmetic and geometric means:

\[
\frac{1}{M} \sum_{k=1}^{M} x_k \geq \left( \prod_{k=1}^{M} x_k \right)^{1/M}
\]

which is so easily obtained in the present framework.
Section 4.9 Complements 185

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 \( \theta \), and let

\[
g_i(\theta) = f(\theta^i) - E_{z|y,\theta^i}\left\{ \ln \left( \frac{p(y, z, \theta)}{p(y, z, \theta^i)} \right) \right\}
\]  (4.9.39)

Clearly \( g_i(\theta) \) satisfies:

\[
g_i(\theta^i) = f(\theta^i)
\]  (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:

\[
g_i(\theta) \geq f(\theta^i) - \ln \left[ E_{z|y,\theta^i} \left\{ \frac{p(y, z, \theta)}{p(y, z, \theta^i)} \right\} \right]
\]

\[
= f(\theta^i) - \ln \left[ E_{z|y,\theta^i} \left\{ \frac{p(y, z, \theta)}{p(z|y, \theta^i)p(y, \theta^i)} \right\} \right]
\]

\[
= f(\theta^i) - \ln \left[ \frac{1}{p(y, \theta^i)} \int p(y, z, \theta) \, dz \right]
\]

\[
= f(\theta^i) + \ln \left[ \frac{p(y, \theta)}{p(y, \theta^i)} \right]
\]

\[
= f(\theta^i) + \left[ f(\theta) - f(\theta^i) \right] = f(\theta)
\]  (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:

\[
\text{Expectation-Maximization (EM) Algorithm}
\]

\[
\theta^0 \text{ = given}
\]

For \( i = 0, 1, 2, \ldots \):

* **Expectation step:** Evaluate
  \[ E_{z|y,\theta^i}\{\ln p(y, z, \theta)\} \equiv g_i(\theta) \]

* **Maximisation step:** Compute
  \[ \theta^{i+1} = \arg \max_{\theta} g_i(\theta) \]

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) may also
be rather challenging. Somewhat paradoxically, these difficulties associated with the EM algorithm may have 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 which lie in a pre-specified 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):

\[ y(t) = \sum_{k=1}^{\tilde{n}} \beta_k e^{i\omega_k t} + e(t); \quad \beta_k = \alpha_k e^{i\varphi_k}, \quad t = 0, \ldots, N-1 \quad (4.9.43) \]

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 \( \tilde{n} \) takes on values close to \( N \) or when \( \tilde{n} \ll N \) 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 pre-specified frequency band comprised by the following Fourier frequencies:

\[ \left\{ \frac{2\pi}{N} k_1, \frac{2\pi}{N} k_2, \ldots, \frac{2\pi}{N} k_M \right\} \quad (4.9.44) \]

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

\[ n \leq \tilde{n} \quad (4.9.45) \]

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 in two parts. From a theoretical standpoint, estimating \( n < \tilde{n} \)...
components of (4.9.43) (in the presence of the remaining components and noise) cannot produce more accurate estimates than estimating all \( \hat{n} \) 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 may give better performance than a non-frequency-selective counterpart that deals with all \( \hat{n} \) components of (4.9.43). This is so because some components of (4.9.43) that do not belong to (4.9.44) may not be well-described by a sinusoidal model; consequently, treating such components as interference and eliminating them from the model may 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 above two desirable features. The following notation will be frequently used in the following:

\[
\begin{align*}
\omega_k &= e^{j\frac{2\pi k}{N}} \quad k = 0, 1, \ldots, N - 1 \\
u_k &= [w_k, \ldots, w_N^m]^T \\
v_k &= [1, w_k, \ldots, w_N^{N-1}]^T \\
y &= [y(0), \ldots, y(N - 1)]^T \\
Y_k &= v_k^* y_k \\
e &= [e(0), \ldots, e(N - 1)]^T \\
E_k &= v_k^* e_k \\
\omega_k &= [e^{j\omega_k}, \ldots, e^{jm\omega_k}]^T \\
b(\omega_k) &= [1, e^{j\omega_k}, \ldots, e^{j(N-1)\omega_k}]^T
\end{align*}
\]

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:

\[
u_k Y_k = \begin{bmatrix} a(\omega_1), \ldots, a(\omega_n) \end{bmatrix} \begin{bmatrix} \beta_1 v_k^* b(\omega_1) \\ \vdots \\ \beta_n v_k^* b(\omega_n) \end{bmatrix} + \Gamma u_k + u_k E_k \tag{4.9.55}
\]

where \( \Gamma \) is an \( m \times m \) matrix defined in equation (4.9.61) below (as will become clear shortly, the definition of \( \Gamma \) has no importance for what follows, and hence it is not repeated here).

To prove (4.9.55), we first write the data vector \( y \) as

\[
y = \sum_{\ell=1}^{n} \beta_\ell b(\omega_\ell) + e \tag{4.9.56}
\]
Next, we note that (for \( p = 1, \ldots, m \)):

\[
\begin{align*}
  w^p_k [v^*_k b(\omega)] &= \sum_{t=0}^{N-1} e^{i(\omega - \frac{2\pi}{N} k)t} e^{\frac{2\pi}{N} kp} \\
&= e^{i\omega p} \sum_{t=0}^{N-1} e^{i(\omega - \frac{2\pi}{N} k)(t-p)} \\
&= e^{i\omega p} [v^*_k b(\omega)] + e^{i\omega p} \left[ \sum_{t=0}^{p-1} e^{i\omega(t-p)} e^{-i\frac{2\pi}{N} k(t-p)} - \sum_{t=N}^{N+p-1} e^{i\omega(t-p)} e^{-i\frac{2\pi}{N} k(t-p)} \right] \\
&= e^{i\omega p} [v^*_k b(\omega)] + e^{i\omega p} \sum_{t=1}^{p} \left[ e^{-i\omega t} e^{i\frac{2\pi}{N} k t} - e^{i\omega(N-t)} e^{i\frac{2\pi}{N} k t} \right] \\
&= e^{i\omega p} [v^*_k b(\omega)] + \sum_{\ell=1}^{p} e^{i\omega(p-\ell)} (1 - e^{i\omega N}) w^\ell_k
\end{align*}
\]  

(4.9.57)

Let (for \( p = 1, \ldots, m \)):

\[
\gamma^*_p(\omega) = (1 - e^{i\omega N}) \left[ e^{i\omega(p-1)}, e^{i\omega(p-2)}, \ldots, e^{i\omega}, 1, 0, \ldots, 0 \right] \quad (1 \times m)
\]  

(4.9.58)

Using (4.9.58) we can rewrite (4.9.57) in the following more compact form (for \( p = 1, \ldots, m \)):

\[
\begin{align*}
  w^p_k [v^*_k b(\omega)] &= e^{i\omega p} [v^*_k b(\omega)] + \gamma^*_p(\omega) u_k \\
\end{align*}
\]  

(4.9.59)

or, equivalently,

\[
\begin{align*}
  u_k [v^*_k b(\omega)] &= a(\omega) [v^*_k b(\omega)] + \begin{bmatrix} \gamma^*_1(\omega) \\ \vdots \\ \gamma^*_n(\omega) \end{bmatrix} u_k \\
\end{align*}
\]  

(4.9.60)

From (4.9.56) and (4.9.60) it follows that

\[
\begin{align*}
  u_k Y_k &= \sum_{\ell=1}^{\hat{n}} \beta_{\ell} u_k [v^*_k b(\omega_{\ell})] + u_k E_k \\
  &= [a(\omega_1), \ldots, a(\omega_{\hat{n}})] \begin{bmatrix} \beta_{1} v^*_k b(\omega_1) \\ \vdots \\ \beta_{\hat{n}} v^*_k b(\omega_{\hat{n}}) \end{bmatrix} + \left\{ \sum_{\ell=1}^{\hat{n}} \beta_{\ell} \begin{bmatrix} \gamma^*_1(\omega_{\ell}) \\ \vdots \\ \gamma^*_n(\omega_{\ell}) \end{bmatrix} \right\} u_k + u_k E_k \\
  &= \begin{bmatrix} a(\omega_1) \\ \vdots \\ a(\omega_{\hat{n}}) \end{bmatrix} \begin{bmatrix} \beta_{1} v^*_k b(\omega_1) \\ \vdots \\ \beta_{\hat{n}} v^*_k b(\omega_{\hat{n}}) \end{bmatrix} + \left\{ \sum_{\ell=1}^{\hat{n}} \beta_{\ell} \begin{bmatrix} \gamma^*_1(\omega_{\ell}) \\ \vdots \\ \gamma^*_n(\omega_{\ell}) \end{bmatrix} \right\} u_k + u_k E_k
\end{align*}
\]  

(4.9.61)

which proves (4.9.55).

In the following 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

\[ A = [a(\omega_1), \ldots, a(\omega_n)] \]  \hspace{1cm} (4.9.62)

\[ x_k = \begin{bmatrix} \beta_1 v_k^* b(\omega_1) \\ \vdots \\ \beta_n v_k^* b(\omega_n) \end{bmatrix} \] \hspace{1cm} (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 following additional notation:

\[ Y = [u_{k_1} Y_{k_1}, \ldots, u_{k_M} Y_{k_M}], \quad (m \times M) \] \hspace{1cm} (4.9.64)

\[ E = [u_{k_1} E_{k_1}, \ldots, u_{k_M} E_{k_M}], \quad (m \times M) \] \hspace{1cm} (4.9.65)

\[ U = [u_{k_1}, \ldots, u_{k_M}], \quad (m \times M) \] \hspace{1cm} (4.9.66)

\[ X = [x_{k_1}, \ldots, x_{k_M}], \quad (n \times M) \] \hspace{1cm} (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:

\[ Y = AX + U + \tilde{A} \tilde{X} + E \] \hspace{1cm} (4.9.68)

Next we assume that

\[ M \geq n + m \] \hspace{1cm} (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):

\[ \Pi_U^+ = I - U^* (UU^*)^{-1} U \] \hspace{1cm} (4.9.70)

We will eliminate the second term in (4.9.68) by post-multiplying (4.9.68) with \( \Pi_U^+ \) (see below). 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 \( E_k = O(N^{1/2}) \) (stochastically), whereas the order of the elements of \( X \) is typically \( O(N) \).

(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) above does not hold, let us consider a generic out-of-band component \( (\omega, \beta) \). 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 \( \beta u_k [v_k^* b(\omega)] \) and, consequently, the part of \( Y \) due to this generic component is

\[
\begin{bmatrix}
v_{k_1}^* b(\omega) \\
0 & \ddots \\
0 & v_{k_M}^* b(\omega)
\end{bmatrix}
\]

(4.9.71)

Even if \( \omega \) 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 may 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 post-multiplication of it by \( \Pi_{\bar{U}}^\perp \) (see below). A similar argument shows that the noise term in (4.9.68) is also attenuated by post-multiplying (4.9.68) with \( \Pi_{\bar{U}}^\perp \).

It follows from the above discussion and (4.9.68) that

\[
Y \Pi_{\bar{U}}^\perp \simeq AX \Pi_{\bar{U}}^\perp
\]

(4.9.72)

This equation resembles equation (4.7.7) on which the standard ESPRIT method is based, provided that

\[
\text{rank} \left( X \Pi_{\bar{U}}^\perp \right) = n \quad (4.9.73)
\]

(similarly to \( \text{rank}(C) = n \) 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 \( k = 1, \ldots, n \)).

To prove (4.9.73) we first note that \( \text{rank}(\Pi_{\bar{U}}^\perp) = M - m \), 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

\[
\text{rank} \left( \begin{bmatrix} X \\ U \end{bmatrix} \right) = m + n \quad (4.9.74)
\]

To verify this equivalence let us decompose \( X \) additively as follows:

\[
X = X \Pi_{U} + X \Pi_{\bar{U}} = X U^* (U U^*)^{-1} U + X V^* V
\]

(4.9.75)

where the \( M \times (M - m) \) matrix \( V^* \) comprises a unitary basis of \( N(U) \); hence, \( U V^* = 0 \) and \( V V^* = I \). Now, the matrix in (4.9.74) has the same rank as

\[
\begin{bmatrix}
I \\
0 & -U U^* (U U^*)^{-1}
\end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} = \begin{bmatrix} XV^* V \\ U \end{bmatrix}
\]

(4.9.76)

(we used (4.9.75) to obtain (4.9.76)), which, in turn, has the same rank as

\[
\begin{bmatrix} XV^* V \\ U \end{bmatrix} \begin{bmatrix} [V V^* X^* U^*] \\
V V^* X^*
\end{bmatrix} = \begin{bmatrix} XV^* V X^* & 0 \\
0 & U U^*
\end{bmatrix}
\]

(4.9.77)

However, \( \text{rank}(U U^*) = m \). Hence, (4.9.74) holds if and only if

\[
\text{rank}(X V^* V X^*) = n
\]
As
\[ \text{rank}(X^*V^*X^*) = \text{rank}(X^*P_1^TX^*) = \text{rank}(X^*P_1^T) \]
the equivalence between (4.9.73) and (4.9.74) is proven.

It follows from the equivalence shown above and the definition of \( X \) and \( U \) that we want to prove that
\[
\begin{align*}
\text{rank} & \begin{pmatrix} v_{k_1}^* b(\omega_1) & \cdots & v_{k_M}^* b(\omega_1) \\
\vdots & & \vdots \\
v_{k_1}^* b(\omega_n) & \cdots & v_{k_M}^* b(\omega_n) \\
u_{k_1} & \cdots & u_{k_M} \\
\end{pmatrix} \\
(2n \times M)
\end{align*}
\]

As
\[
v_{k}^* b(\omega) = \sum_{t=0}^{N-1} e^{i(\omega - \frac{2\pi k}{M})} = \frac{1 - e^{iN(\omega - \frac{2\pi k}{M})}}{1 - e^{i(\omega - \frac{2\pi k}{M})}} = \frac{1 - e^{iN\omega}}{w_k - e^{i\omega}}
\]
we can rewrite the matrix in (4.9.78) as follows:
\[
\begin{pmatrix}
1 - e^{iN\omega_1} & \cdots & 0 \\
\vdots & & \vdots \\
0 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
w_{k_1} & \cdots & w_{k_M} \\
w_{k_1} - e^{i\omega_1} & \cdots & w_{k_M} - e^{i\omega_1} \\
w_{k_1} - e^{i\omega_n} & \cdots & w_{k_M} - e^{i\omega_n} \\
w_{k_1} & \cdots & w_{k_M} \\
0 & \cdots & 1
\end{pmatrix}
\]

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
\[
\frac{\rho_1 z}{z - e^{i\omega_1}} + \cdots + \frac{\rho_n z}{z - e^{i\omega_n}} + \rho_{n+1} z + \cdots + \rho_{n+m} z^m
\]
\[
= z \left( \frac{\rho_1}{z - e^{i\omega_1}} + \cdots + \frac{\rho_n}{z - e^{i\omega_n}} + \rho_{n+1} + \cdots + \rho_{n+m} z^{m-1} \right)
\]
is equal to zero at \( z = w_{k_1}, \ldots, z = w_{k_M} \). However, (4.9.80) can only have \( m + n - 1 \leq M \) zeroes of the above form. 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
\[
m \geq n
\]
Chapter 4 Parametric Methods for Line Spectra

(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 $Y\Pi\hat{U}$ is $n$, and that

$$\hat{S} \simeq A\hat{C}$$

(4.9.82)

where $\hat{C}$ is an $n \times n$ nonsingular transformation matrix, and

$$\hat{S} = \text{the } m \times n \text{ matrix whose columns are the left singular vectors of } Y\Pi\hat{U} \text{ associated with the } n \text{ largest singular values.}$$

(4.9.83)

Equation (4.9.82) is very similar to (4.7.7), and hence it can be used in an ESPRIT-like approach to estimate the frequencies $\{\omega_k\}_{k=1}^n$. Following 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 $M/2$:

$$m = \lfloor M/2 \rfloor$$

(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 the above 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 $Y\Pi\hat{U}$ in (4.9.72) is $m \times M$ and its rank is generically equal to

$$\min\{\text{rank}(Y), \text{rank}(\Pi\hat{U})\} = \min(m, M - m)$$

(4.9.85)

Evidently the above 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, interestingly, the proposed FRES-ESPRIT method with $M = N$ is equivalent to the standard ESPRIT method. For $M = N$ we have that

$$[b_1, \ldots, b_N] \triangleq \begin{bmatrix} w_1 & \cdots & w_N \\ w_1^2 & \cdots & w_N^2 \\ \vdots & \ddots & \vdots \\ w_1^N & \cdots & w_N^N \end{bmatrix} = \begin{bmatrix} U \\ \bar{U} \end{bmatrix}_{N \times m} \{ m \} N - m$$

(4.9.86)

where $U$ is as defined before (with $M = N$) and $\bar{U}$ is defined via (4.9.86). Note that:

$$UU^* = NI; \quad \bar{U}\bar{U}^* = NI; \quad U\bar{U}^* = 0; \quad U^*U + \bar{U}^*\bar{U} = NI$$

(4.9.87)
Hence
\[ \Pi^I_U = I - \frac{1}{N} U^* U = \frac{1}{N} \bar{U}^* \bar{U} \]  \hfill (4.9.88)

Also, note that (for \( p = 1, \ldots, m \)):

\[
w_k^p y_k = \sum_{t=0}^{N-1} y(t) e^{-j \frac{2\pi}{N} k(t-p)}
\]

\[
= \sum_{t=0}^{p-1} y(t) w_k^{p-t} + \sum_{t=p}^{N-1} y(t) w_k^{N+p-t}
\]

\[
= [y(p-1), \ldots, y(0), 0, \ldots, 0] \begin{bmatrix} w_k \\ \vdots \\ w_k^m \end{bmatrix} + [0, \ldots, 0, y(N-1), \ldots, y(p)] \begin{bmatrix} w_k \\ \vdots \\ w_k^N \end{bmatrix}
\]

\[
\triangleq \mu_p^* u_k + \psi_p^* b_k \quad \hfill (4.9.89)
\]

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 \( Y^I_U \) used in the FRES–ESPRIT method can be written as (cf. (4.9.86)–(4.9.89)):

\[
[ u_1 Y_1, \ldots, u_N Y_N ] \Pi^I_U = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_m \end{bmatrix} [ u_1, \ldots, u_N ] + \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_m^* \end{bmatrix} [ b_1, \ldots, b_N ] \bar{U}^* \bar{U} \cdot \frac{1}{N}
\]

\[
\begin{bmatrix} \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_m \end{bmatrix} U \\ \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_m^* \end{bmatrix} 0 \end{bmatrix} \bar{U}^* \bar{U} \cdot \frac{1}{N}
\]

\[
= \begin{bmatrix} y(N-m) & \cdots & y(1) \\ y(N-m+1) & \cdots & y(2) \\ \vdots & \vdots & \vdots \\ y(N-1) & \cdots & y(m) \end{bmatrix} \bar{U} \quad \hfill (4.9.90)
\]

It follows from (4.9.90) that the \( n \) principal (or dominant) left singular vectors of \( Y^I_U \) are equal to the \( n \) principal eigenvectors of the following matrix (obtained by post-multiplying the right-hand side of (4.9.90) with its conjugate transpose and using the fact that \( U \bar{U}^* = NI \) from (4.9.87)):

\[
\begin{bmatrix} y(N-m) & \cdots & y(1) \\ \vdots & \vdots & \vdots \\ y(N-1) & \cdots & y(m) \end{bmatrix} \begin{bmatrix} y^*(N-m) & \cdots & y^*(N-1) \\ \vdots & \vdots & \vdots \\ y^*(1) & \cdots & y^*(m) \end{bmatrix}
\]

\[
= \sum_{t=1}^{N-m} \begin{bmatrix} y(t) \\ \vdots \\ y(t+m-1) \end{bmatrix} \begin{bmatrix} y^*(t), \ldots, y^*(t+m-1) \end{bmatrix} \quad \hfill (4.9.91)
\]
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,

\[
y(t) = \sum_{k=1}^{n} \beta_k e^{i \omega_k t}, \quad t = 0, 1, 2, \ldots
\]  

(4.9.92)

a data vector of length \( m \) can be written as

\[
\begin{bmatrix}
y(0) \\
y(1) \\
\vdots \\
y(m-1)
\end{bmatrix} =
\begin{bmatrix}
1 & \cdots & 1 \\
e^{i \omega_1} & \cdots & e^{i \omega_n} \\
\vdots & \cdots & \vdots \\
e^{i(m-1) \omega_1} & \cdots & e^{i(m-1) \omega_n}
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\vdots \\
\beta_n
\end{bmatrix} \triangleq A \beta
\]  

(4.9.93)

The matrix \( A \) introduced above 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 following discussion 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 uniquely defined via the equation:

\[
1 + c_1 z + \cdots + c_n z^n = \prod_{k=1}^{n} (1 - z e^{-i \omega_k})
\]  

(4.9.94)

Then, it can be readily checked (see (4.5.21)) that the matrix

\[
C^* = \begin{bmatrix}
1 & c_1 & \cdots & c_n \\
c_{n-1} & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \vdots \\
0 & \cdots & 1 & c_n
\end{bmatrix}, \quad (m-n) \times m
\]  

(4.9.95)

satisfies

\[
C^* A = 0
\]  

(4.9.96)

(to verify (4.9.96) it is enough to observe from (4.9.94) that \( 1 + c_1 e^{i \omega_k} + \cdots + c_n e^{i n \omega_k} = 0 \) for \( k = 1, \ldots, n \)). Furthermore, as \( \text{rank}(C) = m-n \) and \( \dim(\mathcal{N}(A^*)) = m - n \) too, it follows from (4.9.96) that

\[
C \text{ is a basis for the null space of } A^*, \mathcal{N}(A^*)
\]  

(4.9.97)

The matrix \( C \) plays an important role in the derivation and analysis of several frequency estimators, see, \textit{e.g.}, Section 4.5, [Bresler and Macovski 1986], and [Stoica and Sharam 1990].
Section 4.9  Complements  195

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 equation (compare with (4.9.92)):

$$y(t, \tilde{t}) = \sum_{k=1}^{n} \beta_k e^{i\omega_k t} e^{i\omega_k \tilde{t}}, \quad t, \tilde{t} = 0, 1, 2, \ldots$$ (4.9.98)

Let

$$\gamma_k = e^{i\omega_k}, \quad \lambda_k = e^{i\tilde{\omega}_k}$$ (4.9.99)

Using this notation allows us to write (4.9.98) in a more compact form,

$$y(t, \tilde{t}) = \sum_{k=1}^{n} \beta_k \gamma_k^t \lambda_k^\tilde{t}$$ (4.9.100)

Moreover, equation (4.9.100) (unlike (4.9.98)) also covers the case of damped (2D) sinusoidal signals, for which

$$\gamma_k = e^{\mu_k + i\omega_k}, \quad \lambda_k = e^{\tilde{\mu}_k + i\tilde{\omega}_k}$$ (4.9.101)

with \(\{\mu_k, \tilde{\mu}_k\}\) being the damping parameters (\(\mu_k, \tilde{\mu}_k \leq 0\)).

The following notation will be frequently used in this complement:

$$g_i^* = [\gamma_1^i \ldots \gamma_n^i]$$ (4.9.102)

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 \\ \vdots & \ddots \\ 0 & \gamma_n \end{bmatrix}$$ (4.9.103)

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ \vdots & \ddots \\ 0 & \lambda_n \end{bmatrix}$$ (4.9.104)

$$\beta = [\beta_1 \ldots \beta_n]^T$$ (4.9.105)

$$A_L = \begin{bmatrix} 1 & \ldots & 1 \\ \lambda_1 & \ldots & \lambda_n \\ \vdots & \ddots & \vdots \\ \lambda_{L-1} & \ldots & \lambda_{L-1} \end{bmatrix}$$ for \(L \geq n\) (4.9.106)

Using (4.9.102), (4.9.104), and (4.9.105) we can write:

$$y(t, \tilde{t}) = g_i^* \Lambda^T \beta$$ (4.9.107)
Chapter 4  Parametric Methods for Line Spectra

Hence, similarly to (4.9.93), we can write the $m\bar{m} \times 1$ data vector obtained from (4.9.98) for $t = 0, \ldots, m - 1$ and $\bar{t} = 0, \ldots, \bar{m} - 1$ as:

$$
\begin{bmatrix}
y(0, 0) \\
y(0, \bar{m} - 1) \\
\vdots \\
y(m - 1, 0) \\
y(m - 1, \bar{m} - 1)
\end{bmatrix} =
\begin{bmatrix}
g_0^0 \Lambda^0 \\
g_0^\bar{m} \Lambda^{\bar{m} - 1} \\
\vdots \\
g_{m - 1}^0 \Lambda^0 \\
g_{m - 1}^{\bar{m} - 1} \Lambda^{\bar{m} - 1}
\end{bmatrix} \beta \triangleq \mathcal{A} \beta
$$

(4.9.108)

The matrix $\mathcal{A}$ defined above, i.e.,

$$
\mathcal{A} =
\begin{bmatrix}
g_0^0 \Lambda^0 \\
g_0^\bar{m} \Lambda^{\bar{m} - 1} \\
\vdots \\
g_{m - 1}^0 \Lambda^0 \\
g_{m - 1}^{\bar{m} - 1} \Lambda^{\bar{m} - 1}
\end{bmatrix}, \quad (m \bar{m} \times n)
$$

(4.9.109)

plays the same role for 2D sinusoidal signals as the matrix $A$ in (4.9.93) 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:

$$
y(t, \bar{t}) = [\lambda_1^t \quad \ldots \quad \lambda_n^t] \Gamma^t \beta
$$

(4.9.110)

This means that $\mathcal{A}$ can also be written as follows:

$$
\mathcal{A} =
\begin{bmatrix}
A_0 \Gamma^0 \\
\vdots \\
A_{m - 1} \Gamma^{m - 1}
\end{bmatrix}
$$

(4.9.111)

Similarly to (4.9.94), let us define the parameters $\{c_k\}_{k=1}^n$ uniquely via the equation

$$
1 + c_1 z + \cdots + c_n z^n = \prod_{k=1}^n \left( 1 - \frac{z}{\lambda_k} \right)
$$

(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 \( m > n \):

\[
C^* = \begin{bmatrix}
1 & c_1 & \cdots & c_n & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & 1 & \cdots & c_n & 0 \\
0 & 0 & \cdots & 1 & 0
\end{bmatrix}, \quad (\tilde{m} - n) \times \tilde{m}
\]

(4.9.113)

and note that (cf. (4.9.96))

\[
C^* A_{\tilde{m}} = 0
\]

(4.9.114)

It follows from (4.9.111) and (4.9.114) that

\[
\begin{bmatrix}
C^* \\
0 \\
\vdots \\
0
\end{bmatrix} A = 0
\]

(4.9.115)

Hence, we have found \((m\tilde{m} - mn)\) vectors of the sought basis for \( \mathcal{N}(A^*) \). It remains to find \((m-1)n\) additional (linearly independent) vectors of this basis (note that \( \dim(\mathcal{N}(A^*)) = m\tilde{m} - n \)). To find the remaining vectors we need an approach which is rather different from that used so far.

Let us assume that

\[
\lambda_k \neq \lambda_p \quad \text{for} \quad k \neq p
\]

(4.9.116)

and let the vector

\[
b^* = [b_1, \ldots, b_n]
\]

be defined via the linear (interpolation) equation

\[
b^* A_n = [\gamma_1, \ldots, \gamma_n]
\]

(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 \( \{ \gamma_k \} \), and 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 obtain easily

\[
b^* A_n \Gamma^t = [\gamma_1, \ldots, \gamma_n] \Gamma^t = g_{t+1}^*
\]

and hence (see also (4.9.109) and (4.9.111))

\[
b^* \begin{bmatrix}
g_t^* \Lambda^0 \\
g_t^* \Lambda^1 \\
\vdots \\
g_t^* \Lambda^{n-1}
\end{bmatrix} = b^* A_n \Gamma^t = g_{t+1}^* \Lambda^0
\]

(4.9.118)
Next, we assume that
\[ \tilde{m} \geq 2n - 1 \]  
(4.9.119)
which is a weak condition (typically we have \( m, \tilde{m} \gg n \)). Under (4.9.119) we can write (making use of (4.9.118)):
\[
\begin{bmatrix} b^* & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & b^* & 0 \end{bmatrix} \begin{bmatrix} g_t^0 A^0 \\ \vdots \\ g_t^\tilde{m} A^{\tilde{m}-1} \end{bmatrix} - \begin{bmatrix} g_{t+1}^0 A^0 \\ \vdots \\ g_{t+1}^{n-1} A^{n-1} \end{bmatrix} = 0 \]  
(4.9.120)

where
\[
B^* = \begin{bmatrix} b_1 & b_2 & \cdots & b_n & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & b_1 & b_2 & \cdots & b_n & 0 \end{bmatrix} \quad (n \times \tilde{m})
\]

Note that, indeed, we need \( \tilde{m} \geq 2n - 1 \) to be able to write (4.9.120) (if \( \tilde{m} > 2n - 1 \) then the rightmost \( \tilde{m} - 2n - 1 \) 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 \( A \):

\[
\begin{bmatrix} D & I \\ D & I & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & D & I & C^* \end{bmatrix}^{\text{block rows}} \]  
(4.9.121)

where
\[
D = \begin{bmatrix} 1 & c_1 & \cdots & c_n & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & c_1 & \cdots & c_n \\ b_1 & b_2 & \cdots & b_n & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & b_1 & b_2 & \cdots & b_n \end{bmatrix} \quad (\tilde{m} \times \tilde{m})
\]
\[
I = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ -1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & -1 & 0 & \cdots & 0 \end{bmatrix} \quad (\tilde{m} \times \tilde{m})
\]

The matrix in (4.9.121) is of dimension \([ (m-1)\tilde{m} + (\tilde{m} - n) ] \times m\tilde{m} \), that is \((m\tilde{m} - n) \times m\tilde{m} \), and its rank is equal to \( m\tilde{m} - n \) (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 \( A \). 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 doing so the linear dependence of (4.9.121) on the unknown parameters comes 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 be easily extended to the case of 2D signals 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 above 2D basis result depends on the condition (4.9.116); if (4.9.116) is even approximately violated (i.e., if there exist \( \lambda_k \) and \( \lambda_p \) with \( k \neq p \) such that \( \lambda_k \approx \lambda_p \)) then the mapping \( \{ \gamma_k \} \leftrightarrow \{ b_k \} \) may become ill-conditioned, which may result in 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 \( \{ \omega_k, \tilde{\omega}_k \} \) 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 may 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 \( \mathcal{N}(A^*) \).

## 4.10 EXERCISES

**Exercise 4.1: Speed Measurement by a Doppler Radar as a Frequency Determination 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 \( \alpha e^{j\omega t} \) 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:

\[
s(t) = \beta e^{j(\omega - \omega^D)t} + e(t)
\]

(4.10.1)

where \( e(t) \) is measurement noise, \( \omega^D \) is the so-called Doppler frequency,

\[
\omega^D \triangleq 2\omega v / c
\]

and

\[
\beta = \mu a e^{-2i\omega r / c}
\]

Here \( c \) denotes the speed of wave propagation, \( r \) is the object range, and \( \mu \) 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.
which coincides with (5.6.5). Thus, the proof of (5.6.4) is concluded.

**Remark:** The reader may wonder what happens with the formulas derived above 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 GS formula above 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 \( \{ \hat{a}_k \} \), of course).

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 \( \hat{\phi}_{AR}(\omega) \). Similar to the interpretation in Section 5.4.2, the previous observation can be used to intuitively understand 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: \( \{ \hat{\rho}(s) \} \) 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, e.g., [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 then \( \{ \hat{\mu}(s) \} \) 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))

\[
\hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} w_{p,t} y(t)e^{-i\omega t} \right|^2
\]  

(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, as well as 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

\[ z(\omega) = \begin{bmatrix} y(1)e^{-i\omega} \\ \vdots \\ y(N)e^{-iN\omega} \end{bmatrix}, \quad (N \times 1) \]

\[ W = \begin{bmatrix} w_{1,1} & \cdots & w_{1,N} \\ \vdots & \ddots & \vdots \\ w_{K,1} & \cdots & w_{K,N} \end{bmatrix}, \quad (K \times N) \]

and let \([x]_p\) denote the \(p\)th element of a vector \(x\). Using this notation we can rewrite (5.6.10) in the desired form:

\[ \hat{\phi}(\omega) = \frac{1}{K} \sum_{p=1}^{K} ||Wz(\omega)||_p^2 \]

or

\[ \hat{\phi}(\omega) = \frac{1}{K} z^*(\omega)W^*Wz(\omega) \]

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)):

\[ \hat{\phi}_D(\omega) = \frac{1}{2J+1} \sum_{j=-J}^{J} \hat{\phi}_p \left( \omega + j\frac{2\pi}{N} \right) \]

(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

\[ a_j^* = \left[ e^{-i\frac{2\pi}{N}j}, e^{-i\frac{2\pi}{N}(2j)}, \ldots, e^{-i\frac{2\pi}{N}(Nj)} \right] \]

(5.6.13)
we can write

\[
\hat{\phi}_p \left( \omega + j \frac{2\pi}{N} \right) = \frac{1}{N} \left| \sum_{t=1}^{N} y(t)e^{-i\omega t}e^{-i\frac{2\pi}{N}(jt)} \right|^2
\]

\[
= \frac{1}{N} \left| a_j^* z(\omega) \right|^2 = \frac{1}{N} z^*(\omega) a_j a_j^* z(\omega)
\]

which implies that

\[
\hat{\phi}_D(\omega) = \frac{1}{N(2J+1)} z^*(\omega) W_D^* W_D z(\omega)
\]

where

\[
W_D = [a_{-J}, \ldots, a_0, \ldots, a_J]^*, \quad (2J + 1) \times N
\]

This establishes the fact that the Daniell periodogram can be interpreted as a multiwindow method using \(K = 2J + 1\) tapers given by (5.6.16). Similarly to the tapers used by the seemingly more elaborate RFB approach, the Daniell periodogram tapers can also be motivated 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 \(M = N\):

\[
\hat{\phi}_{BT}(\omega) = \sum_{k=-(N-1)}^{N-1} w(k) \hat{r}(k)e^{-i\omega k}
\]

A commonly-used class of windows, including the Hanning and Hamming windows in Table 2.1, is described by the equation:

\[
w(k) = \alpha + \beta \cos(\Delta k) = \left( \alpha + \frac{\beta}{\pi} e^{i\Delta k} + \frac{\beta}{\pi} e^{-i\Delta k} \right)
\]

for various parameters \(\alpha\), \(\beta\), and \(\Delta\). Inserting (5.6.18) into (5.6.17) yields:

\[
\hat{\phi}_{BT}(\omega) = \sum_{k=-(N-1)}^{N-1} \left( \alpha + \frac{\beta}{\pi} e^{i\Delta k} + \frac{\beta}{\pi} e^{-i\Delta k} \right) \hat{r}(k)e^{-i\omega k}
\]

\[
= \alpha \hat{\phi}_p(\omega) + \frac{\beta}{\pi} \hat{\phi}_p(\omega - \Delta) + \frac{\beta}{\pi} \hat{\phi}_p(\omega + \Delta)
\]

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) (as well as (5.6.14)–(5.6.16)) allows us to rewrite \(\hat{\phi}_{BT}(\omega)\) in the following form:

\[
\hat{\phi}_{BT}(\omega) = \frac{1}{N} z^*(\omega) W_{BT}^* W_{BT} z(\omega)
\]
Section 5.6 Complements

where

\[ W_{BT} = \left[ \sqrt{\frac{\alpha}{2}} a_{-\Delta}, \sqrt{\alpha} a_0, \sqrt{\frac{\alpha}{2}} a_{\Delta} \right]^*, \quad (3 \times N) \]  

(5.6.21)

for \( \alpha, \beta \geq 0 \) and where \( a_{\Delta} \) is given by (similarly to \( a_j \) in (5.6.13))

\[ a_{\Delta}^* = [e^{-i\Delta}, \ldots, e^{-i\Delta N}] \]

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 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 byproduct 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. Since the Blackman window has a form similar to (5.6.18), \( \hat{\phi}_{BT}(\omega) \) using the Blackman window can be similarly 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 which are dealt with in some applications of spectral analysis, such as in magnetic resonance spectroscopy, consist of a sum of \textit{exponentially damped sinusoidal components}, (or damped sinusoids, for short), instead of the pure sinusoids as in (4.1.1). Such signals are described by the equation

\[ y(t) = \sum_{k=1}^{n} \beta_k e^{(\rho_k + i\omega_k)t} + e(t), \quad t = 1, \ldots, N \]  

(5.6.22)

where \( \beta_k \) and \( \omega_k \) are the amplitude and frequency of the \( k \)th component (as in Chapter 4), and \( \rho_k < 0 \) is the so-called damping parameter. The (noise-free) signal in (5.6.22) is \textit{nonstationary} and hence it does not have a power spectral density. However, it possesses an \textit{amplitude spectrum} that is defined as follows:

\[ |\beta(\rho, \omega)| = \begin{cases} |\beta_k|, & \text{for } \omega = \omega_k, \rho = \rho_k \quad (k = 1, \ldots, n) \\ 0, & \text{elsewhere} \end{cases} \]  

(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 \textit{energy spectrum}. Similarly to (5.6.23), we can define the energy spectrum of the damped sinusoidal signal in (5.6.22) as a 2D function of \((\rho, \omega)\) 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 \((\beta, \rho, \omega)\) is given by

\[ \sum_{t=1}^{N} |\beta e^{(\rho t + i\omega t)}|^2 = |\beta|^2 e^{i\rho t} \sum_{t=0}^{N-1} e^{2\rho t} = |\beta|^2 e^{2\rho} \frac{1 - e^{2\rho N}}{1 - e^{2\rho}} \]  

(5.6.24)
It follows from (5.6.24) and the above discussion that the energy spectrum can be expressed as a function of the amplitude spectrum in (5.6.23) via the formula:

\[ E(\rho, \omega) = |\beta(\rho, \omega)|^2 L(\rho) \]  

where

\[ L(\rho) = e^{2\rho} \frac{1 - e^{2\rho N}}{1 - e^{2\rho}} \]  

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 following data vector

\[ \tilde{y}(t) = [y(t), y(t+1), \ldots, y(t+m)] \]  

in lieu of the data vector used in (5.4.2). First we explain why, in the case of damped sinusoidal signals, the use of (5.6.27) is preferable to that of

\[ [y(t), y(t-1), \ldots, y(t-m)]^T \]  

(as is used 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 using the data vector in (5.6.27) is given by:

\[ \tilde{y}_F(t) = h^* \tilde{y}(t) = h^* \begin{bmatrix} y(t) \\ \vdots \\ y(t+m) \end{bmatrix}, \quad t = 1, \ldots, N - m \]  

Hence, when performing the filtering operation as in (5.6.29), we lose \( m \) samples from the end of the data string. Because the SNR of those samples is typically rather low (owing to the damping of the signal components), the data loss is not significant. In contrast, the use of (5.4.2) leads to a loss of \( m \) data samples from the beginning of the data string (since (5.4.2) can be computed for \( t = m+1, \ldots, N \)), where the SNR is higher. Hence, in the case of damped sinusoidal signals we should indeed prefer (5.6.29) to (5.4.2).

Next, we derive Capon-like estimates of the amplitude and energy spectra of (5.6.22). Let

\[ \hat{R} = \frac{1}{N-m} \sum_{t=1}^{N-m} \tilde{y}(t)\tilde{y}^*(t) \]  

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:

\[ \frac{1}{N-m} \sum_{t=1}^{N-m} |\tilde{y}_F(t)|^2 = h^* \hat{R} h \]  

By definition, the Capon filter minimizes (5.6.31) under the constraint that the filter passes, without distortion, a generic damped sinusoid with parameters \((\beta, \rho, \omega)\).
The filter output corresponding to such a generic component is given by
\[
\begin{bmatrix}
\beta e^{(p+i\omega)t} \\
\beta e^{(p+i\omega)(t+1)} \\
\vdots \\
\beta e^{(p+i\omega)(t+m)}
\end{bmatrix}
= \begin{bmatrix}
1 \\
e^{p+i\omega} \\
\vdots \\
e^{(p+i\omega)m}
\end{bmatrix}
\beta e^{(p+i\omega)t}
\] (5.6.32)
Hence, the distortionless filtering constraint can be expressed as
\[
h^*a(p, \omega) = 1
\] (5.6.33)
where
\[
a(p, \omega) = \begin{bmatrix} 1, e^{p+i\omega}, \ldots, e^{(p+i\omega)m} \end{bmatrix}^T
\] (5.6.34)
The minimizer of the quadratic function in (5.6.31) under the linear constraint (5.6.33) is given by the familiar formula (see (5.4.7)–(5.4.8)):
\[
h(p, \omega) = \frac{\hat{R}^{-1}a(p, \omega)}{a^*(p, \omega)\hat{R}^{-1}a(p, \omega)}
\] (5.6.35)
where we have stressed, via notation, the dependence of \(h\) on both \(p\) and \(\omega\).

The output of the filter in (5.6.35) due to a possible (generic) damped sinusoid in the signal with parameters \((\beta, p, \omega)\), is given by (cf. (5.6.32) or (5.6.33)):
\[
h^*(p, \omega)\tilde{y}(t) = e^{(p+i\omega)t} + e^F(t), \quad t = 1, \ldots, N - m
\] (5.6.36)
where \(e^F(t)\) denotes the filter output due to noise and to any other signal components. For given \((p, \omega)\), the least-squares estimate of \(\beta\) in (5.6.36) is (see, e.g., Result R32 in Appendix A):
\[
\hat{\beta}(p, \omega) = \sum_{t=1}^{N-m} h^*(p, \omega)\tilde{y}(t)e^{(p-i\omega)t} = \sum_{t=1}^{N-m} e^{2pt}
\] (5.6.37)
Let \(\hat{L}(p)\) be defined similarly to \(L(p)\) in (5.6.26), but with \(N\) replaced by \(N - m\), and let
\[
\hat{Y}(p, \omega) = \frac{1}{\hat{L}(p)} \sum_{t=1}^{N-m} \tilde{y}(t)e^{(p-i\omega)t}
\] (5.6.38)
It follows from (5.6.37), 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:

\[ |\hat{\beta}(\rho, \omega)| = |h^*(\rho, \omega)\hat{Y}(\rho, \omega)| \]  \hspace{1cm} (5.6.39)

and

\[ \hat{E}(\rho, \omega) = |\hat{\beta}(\rho, \omega)|^2 L(\rho) \]  \hspace{1cm} (5.6.40)

**Remark:** We could have estimated the amplitude, \( \beta \), of a generic component with parameters \((\beta, \rho, \omega)\) directly from the unfiltered data samples \( \{y(t)\}_{t=1}^{N} \). However, the use of the Capon filtered data in (5.6.36) 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 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 (5.6.36).

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 \((\rho, \omega)\) plane, and obtain estimates of \((\rho_k, \omega_k)\) as the locations of the \( n \) largest spectral peaks; estimates of \( \beta_k \) can then be derived from (5.6.37) with \((\rho, \omega)\) replaced by the estimated values of \((\rho_k, \omega_k)\). There is empirical evidence that the use of \( \hat{E}(\rho, \omega) \) in general leads to (slightly) more accurate signal parameter estimates than 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 (5.6.39) and (5.6.40), 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 \( \omega \) (currently considered in the analysis) passes through the filter in a distortionless manner; 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 \( \omega \), \( \{\beta e^{i\omega t}\} \), is given by (assuming forward filtering, as in (5.4.2)):

\[ h^* \begin{bmatrix} e^{i\omega t} \\ e^{i\omega(t-1)} \\ \vdots \\ e^{i\omega(t-m)} \end{bmatrix} \beta = \left( h^* \begin{bmatrix} 1 \\ e^{-i\omega} \\ \vdots \\ e^{-i\omega m} \end{bmatrix} \right) \beta e^{i\omega t} \]  \hspace{1cm} (5.6.41)
Section 5.6 Complements

For backward filtering, as used in Complement 5.6.3, a similar result can be derived. It follows from (5.6.41) that the design objective in (a) above can be expressed mathematically via the following linear constraint on $h$:

$$h^*a(\omega) = 1$$  \hspace{1cm} (5.6.42)

where

$$a(\omega) = [1, e^{-i\omega}, \ldots, e^{-i\omega_m}]^T$$  \hspace{1cm} (5.6.43)

(see (5.4.5)–(5.4.7)). Regarding the second design objective, its statement in (b) above 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 $\omega$, let us choose $h$ such that the filter output, $\{h^*\hat{y}(t)\}$, where

$$\hat{y}(t) = [y(t), y(t-1), \ldots, y(t-m)]^T$$

is as close as possible in a least-squares (LS) sense to a sinusoid with frequency $\omega$ and constant amplitude $\beta$. Mathematically, we obtain both $h$ and $\beta$, for a given $\omega$, by minimizing the LS criterion:

$$\min_{h, \beta} \frac{1}{N-m} \sum_{t=m+1}^{N} |h^*\hat{y}(t) - \beta e^{i\omega t}|^2 \quad \text{subject to } h^*a(\omega) = 1$$  \hspace{1cm} (5.6.44)

Note that the estimation of the amplitude and phase (i.e., $|\beta|$ and $\arg(\beta)$) of the sinusoid with frequency $\omega$ is an intrinsic part of the method based on (5.6.44). This observation motives the name of Amplitude and Phase Estimation (APES) given to the method described by (5.6.44).

Because (5.6.44) is a linearly constrained quadratic problem, we should be able to find its solution in closed form. Let

$$g(\omega) = \frac{1}{N-m} \sum_{t=m+1}^{N} \hat{y}(t)e^{-i\omega t}$$  \hspace{1cm} (5.6.45)

Then, a straightforward calculation shows that the criterion function in (5.6.44) can be rewritten as:

$$\frac{1}{N-m} \sum_{t=m+1}^{N} |h^*\hat{y}(t) - \beta e^{i\omega t}|^2$$

$$= h^*\hat{R}h - \beta^*h^*g(\omega) - \beta g^*(\omega)h + |\beta|^2$$

$$= |\beta - h^*g(\omega)|^2 + h^*\hat{R}h - |h^*g(\omega)|^2$$

$$= |\beta - h^*g(\omega)|^2 + h^*[\hat{R} - g(\omega)g^*(\omega)]h$$  \hspace{1cm} (5.6.46)
Chapter 5 Filter Bank Methods

where

\[ \hat{R} = \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t)\tilde{y}^*(t) \]  

(5.6.47)

(see (5.4.18)). The minimization of (5.6.46) with respect to \( \beta \) is immediate:

\[ \beta(\omega) = h^*g(\omega) \]  

(5.6.48)

Inserting (5.6.48) into (5.6.46) yields the following problem whose solution will determine the filter coefficient vector:

\[ \min_{h} h^*\hat{Q}(\omega)h \quad \text{subject to } h^*a(\omega) = 1 \]  

(5.6.49)

where

\[ \hat{Q}(\omega) = \hat{R} - g(\omega)g^*(\omega) \]  

(5.6.50)

As (5.6.49) has the same form as the Capon filter design problem (see (5.4.7)), the solution to (5.6.49) is readily derived (compare with (5.4.8)):

\[ h(\omega) = \frac{\hat{Q}^{-1}(\omega)a(\omega)}{a^*(\omega)\hat{Q}^{-1}(\omega)a(\omega)} \]  

(5.6.51)

A direct implementation of (5.6.51) 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 (5.6.51) as follows:

\[ \hat{Q}^{-1}(\omega) = \left[ \hat{R} - g(\omega)g^*(\omega) \right]^{-1} = \hat{R}^{-1} + \frac{\hat{R}^{-1}g(\omega)g^*(\omega)\hat{R}^{-1}}{1 - g^*(\omega)\hat{R}^{-1}g(\omega)} \]  

(5.6.52)

Inserting (5.6.52) into (5.6.51) yields the following expression for the APES filter:

\[ h(\omega) = \frac{1 - g^*(\omega)\hat{R}^{-1}g(\omega)}{1 - g^*(\omega)\hat{R}^{-1}g(\omega)} \hat{R}^{-1}a(\omega) + \left[ g^*(\omega)\hat{R}^{-1}a(\omega) \right] \hat{R}^{-1}g(\omega) \]  

(5.6.53)

From (5.6.48) and (5.6.53) 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):

\[ \beta(\omega) = \frac{a^*(\omega)\hat{R}^{-1}g(\omega)}{1 - g^*(\omega)\hat{R}^{-1}g(\omega)} \]  

(5.6.54)

Compared with the **Capon estimate of the amplitude spectrum** given by

\[ \beta(\omega) = \frac{a^*(\omega)\hat{R}^{-1}g(\omega)}{a^*(\omega)\hat{R}^{-1}a(\omega)} \]  

(5.6.55)
we see that the APES estimate in (5.6.54) is more computationally involved, 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, (5.6.53), 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.

The above 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 as well as 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 spectral line analysis, a better method than both Capon and 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 $\{\beta_k\}$ using the APES formula (5.6.54) evaluated at the frequency estimates obtained in the previous step.

The above 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, owing to the failure of a measuring device, or owing to the impossibility to perform 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 lost information due to the missing data samples. The problem with these assumptions, though, is that they are not generally easy to check in applications, either \textit{a priori} or \textit{a posteriori}. The GAPES approach presented here is based on the sole assumption that \textit{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 (5.6.44) 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 $\omega$ of both the APES filter and the amplitude spectrum. Also, we note that in applications the frequency variable is usually sampled as follows:

$$\omega_k = \frac{2\pi}{K} k, \quad k = 1, \ldots, K$$

(5.6.56)

where $K$ is an integer (much) larger than $N$. Making use of the above notation and (5.6.56) we rewrite the APES criterion as follows:

$$\min \sum_{k=1}^{K} \sum_{t=m+1}^{N} \left| h^*(\omega_k) \tilde{y}(t) - \beta(\omega_k) e^{i\omega_k t} \right|^2$$

subject to $h^*(\omega_k) a(\omega_k) = 1$ for $k = 1, \ldots, K$

(5.6.57)

Evidently, the minimization of the criterion in (5.6.57) with respect to $\{h(\omega_k)\}$ and $\{\beta(\omega_k)\}$ reduces to the minimization of the inner sum in (5.6.57) for each $k$. Hence, in the full-data case the problem in (5.6.57) is equivalent to the standard APES problem in equation (5.6.44) in Complement 5.6.4. However, in the gapped data case the form of the APES criterion in (5.6.57) turns out to be more convenient than that in (5.6.44), as we will see below.

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 (5.6.57) 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 $\omega_k$, 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 (5.6.57).
The criterion in (5.6.57) 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 (5.6.57) (see Complement 4.9.5 for a general description of cyclic minimizers). A step-by-step description of GAPES is as follows:

**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 (5.6.57).

**Step 2.** Use the most recent estimate of \( y_u \) to estimate \( \{h(\omega_k)\} \) and \( \{\beta(\omega_k)\} \) via the minimization of (5.6.57).

**Step 3.** Check the convergence of the iteration, e.g., by checking whether the relative change of the criterion between two consecutive iterations is smaller than a pre-assigned 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.

To reduce the computational burden of the above 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.

A cyclic minimizer reduces the criterion function at each iteration (see the discussion 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 (5.6.57) is bounded from below by zero, we can conclude that the GAPES algorithm converges to a minimum point of (5.6.57). 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. The initialization step, as well as the remaining steps in the GAPES algorithm, are discussed in more detail below.

**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 (5.6.47) 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:

\[
\min_{y_u} \sum_{k=1}^{K} \sum_{t=m+1}^{N} \left| \hat{h}(\omega_k) \hat{y}(t) - \hat{\beta}(\omega_k) e^{i\omega_k t} \right|^2
\]  

(5.6.58)

where \( \hat{y}(t) = [y(t), y(t-1), \ldots, y(t-m)]^T \). We will show that the above minimization problem is quadratic in \( y_u \) (for given \( \hat{h}(\omega_k) \) and \( \hat{\beta}(\omega_k) \)), and thus admits a closed-form solution.

Let \( \hat{h}^*(\omega_k) = [h_{0,k}, h_{1,k}, \ldots, h_{m,k}] \) and define

\[
H_k = \begin{bmatrix}
h_{0,k} & h_{1,k} & \cdots & h_{m,k} & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & h_{0,k} & h_{1,k} & \cdots & h_{m,k}
\end{bmatrix}, \quad (N-m) \times N
\]

\[
\mu_k = \hat{\beta}(\omega_k) \begin{bmatrix}
e^{i\omega_k N} \\
\vdots \\
e^{i\omega_k (m+1)}
\end{bmatrix}, \quad (N-m) \times 1
\]

Using this notation we can write the quadratic criterion in (5.6.58) as

\[
\sum_{k=1}^{K} \left\| H_k \begin{bmatrix}
y(N) \\
\vdots \\
y(1)
\end{bmatrix} - \mu_k \right\|^2
\]  

(5.6.59)

Next, we define the matrices \( A_k \) and \( U_k \) via the following equality:

\[
H_k \begin{bmatrix}
y(N) \\
\vdots \\
y(1)
\end{bmatrix} = A_k y_a + U_k y_u
\]  

(5.6.60)

With this notation, the criterion in (5.6.59) becomes:

\[
\sum_{k=1}^{K} \left\| U_k y_u - (\mu_k - A_k y_a) \right\|^2
\]  

(5.6.61)

The minimizer of (5.6.61) with respect to \( y_u \) is readily found to be (see Result R32 in Appendix A):

\[
\hat{y}_u = \left( \sum_{k=1}^{K} U_k^* U_k \right)^{-1} \left( \sum_{k=1}^{K} U_k^* (\mu_k - A_k y_a) \right)
\]  

(5.6.62)

The inverse matrix above 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 in 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 were 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 the above 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, and so forth. 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 extension of some parametric methods for spectral line analysis is 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 \( \{\beta e^{i\omega t}\} \):

\[
\min_{\beta} \sum_{t=1}^{N} |y(t) - \beta e^{i\omega t}|^2 \quad (5.6.63)
\]

The solution to (5.6.63) is readily found to be

\[
\beta(\omega) = \frac{1}{N} \sum_{t=1}^{N} y(t)e^{-i\omega t} \quad (5.6.64)
\]

The squared modulus of (5.6.64) (scaled by \( N \); see Section 5.2) gives the 1D periodogram

\[
\frac{1}{N} \left| \sum_{t=1}^{N} y(t)e^{-i\omega t} \right|^2 \quad (5.6.65)
\]

In the 2D case, let \( \{y(t, \tilde{t})\} \) (for \( t = 1, \ldots, N \) and \( \tilde{t} = 1, \ldots, \tilde{N} \)) denote the available data matrix, and let \( \{\beta e^{i(\omega t + \tilde{\omega} \tilde{t})}\} \) denote a generic 2D sinusoid. The LS fit of the data to the generic sinusoid, that is:

\[
\min_{\beta} \sum_{t=1}^{N} \sum_{\tilde{t}=1}^{\tilde{N}} |y(t, \tilde{t}) - \beta e^{i(\omega t + \tilde{\omega} \tilde{t})}|^2 \iff \min_{\beta} \sum_{t=1}^{N} \sum_{\tilde{t}=1}^{\tilde{N}} |y(t, \tilde{t})e^{-i(\omega t + \tilde{\omega} \tilde{t})} - \beta|^2
\]

(5.6.66)
Chapter 5 Filter Bank Methods

has the following solution:

$$\beta(\omega, \bar{\omega}) = \frac{1}{NN} \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} y(t, \bar{t}) e^{-i(\omega t + \bar{\omega} \bar{t})}$$  \hspace{1cm} (5.6.67)

Similarly to the 1D case, the scaled squared magnitude of (5.6.67) yields the 2D periodogram

$$\frac{1}{NN} \left| \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} y(t, \bar{t}) e^{-i(\omega t + \bar{\omega} \bar{t})} \right|^2$$  \hspace{1cm} (5.6.68)

which can be efficiently computed by means of a 2D FFT algorithm as described below.

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

$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, \ldots, N - 1$$
$$\bar{\omega}_\ell = \frac{2\pi \ell}{\bar{N}}, \quad \ell = 0, \ldots, \bar{N} - 1$$

The 2D FFT algorithm achieves computational efficiency by making use of the 1D FFT described in Section 2.3. Let

$$Y(k, \ell) = \sum_{t=1}^{N} \sum_{\bar{t}=1}^{\bar{N}} y(t, \bar{t}) e^{-i \left( \frac{2\pi k}{N} t + \frac{2\pi \ell}{\bar{N}} \bar{t} \right)}$$

$$= \sum_{t=1}^{N} e^{-i \frac{2\pi k}{N} t} \sum_{\bar{t}=1}^{\bar{N}} y(t, \bar{t}) e^{-i \frac{2\pi \ell}{\bar{N}} \bar{t}}$$  \hspace{1cm} (5.6.69)

$$= \sum_{t=1}^{N} V_t(\ell) e^{-i \frac{2\pi k}{N} t}$$  \hspace{1cm} (5.6.70)

For each \(t = 1, \ldots, N\), the sequence \(\{V_t(\ell)\}_{\ell=0}^{\bar{N}-1}\) defined in (5.6.69) can be efficiently computed 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 (5.6.70) can be efficiently computed 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

$$N \frac{\bar{N}}{2} \log_2 \bar{N} + \frac{N}{2} \log_2 N = \frac{N \bar{N}}{2} \log_2 (N\bar{N}) \text{ flops}$$  \hspace{1cm} (5.6.71)

If \(N\) or \(\bar{N}\) is not a power of two, zero padding can be used.
Section 5.6 Complements

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. (5.6.68)):

\[
\frac{1}{K} \sum_{p=1}^{K} \left| \sum_{t=1}^{N} \sum_{\tilde{t}=1}^{\tilde{N}} w_p(t, \tilde{t}) y(t, \tilde{t}) e^{-i(\omega t + \omega \tilde{t})} \right|^2
\]  

(5.6.72)

where \( \{w_p(t, \tilde{t})\}_{p=1}^{K} \) 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 \( \{w(t, \tilde{t})\} \) is given by

\[
\sum_{t=1}^{N} \sum_{\tilde{t}=1}^{\tilde{N}} w(t, \tilde{t}) e^{-i(\omega t + \omega \tilde{t})}
\]  

(5.6.73)

Let us define the matrices

\[
W = \begin{bmatrix}
w(1, 1) & \cdots & w(1, \tilde{N}) \\
\vdots & \ddots & \vdots \\
w(N, 1) & \cdots & w(N, \tilde{N})
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
e^{-i(\omega \tilde{\omega})} & \cdots & e^{-i(\omega + \tilde{\omega} \tilde{N})} \\
\vdots & \ddots & \vdots \\
e^{-i(\omega N \tilde{\omega})} & \cdots & e^{-i(\omega N + \tilde{\omega} \tilde{N})}
\end{bmatrix}
\]

and let \( \text{vec}(\cdot) \) denote the vectorization operator which stacks the columns of its matrix argument into a single vector. Also, let

\[
a(\omega) = \begin{bmatrix}
e^{-i\omega} \\
\vdots \\
e^{-iN\omega}
\end{bmatrix}, \quad \tilde{a}(\omega) = \begin{bmatrix}
e^{-i\tilde{\omega}} \\
\vdots \\
e^{-i\tilde{N}\tilde{\omega}}
\end{bmatrix}
\]  

(5.6.74)

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 \( m\tilde{m} \times n\tilde{n} \), is an \( m\tilde{m} \times n\tilde{n} \) matrix whose \( (i, j) \) block of size \( m\tilde{m} \times n\tilde{n} \) 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

\[
w = \text{vec}(W)
\]

\[
= [w(1, 1), \ldots, w(N, 1), \ldots, w(1, \tilde{N}), \ldots, w(N, \tilde{N})]^T
\]  

(5.6.75)

and

\[
b(\omega, \tilde{\omega}) = \text{vec}(B)
\]

\[
= [e^{-i(\omega + \tilde{\omega})}, e^{-i(\omega N + \tilde{\omega})}, \ldots, e^{-i(\omega N + \tilde{\omega} \tilde{N})}, \ldots, e^{-i(\omega(\omega + \tilde{\omega} \tilde{N})}]^T
\]

\[
= \tilde{a}(\omega) \otimes a(\omega)
\]  

(5.6.76)
Chapter 5  Filter Bank Methods

(the last equality in (5.6.76) follows from the definition of $\otimes$). Using (5.6.75) and (5.6.76), we can write (5.6.73) as

$$w^* b(\omega, \bar{\omega})$$

where $w(\omega)$ is the frequency response. This expression 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 that now the matrix $\Gamma$ is given by

$$\Gamma_{2D} = \Gamma_{1D} \otimes \Gamma_{1D}$$

where we have used the fact that $(A \otimes B)(C \otimes D) = AC \otimes BD$ for any conformable matrices (see, e.g., [Horn and Johnson 1985]). Hence,

$$\Gamma_{2D} = \Gamma_{1D} \otimes \Gamma_{1D}$$

where

$$\Gamma_{1D} = \frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} a(\omega) a^*(\omega) d\omega, \quad \Gamma_{1D} = \frac{1}{2\pi} \int_{-\beta\pi}^{\beta\pi} \tilde{a}(\bar{\omega}) \tilde{a}^*(\bar{\omega}) d\omega$$

The above Kronecker product expression of $\Gamma_{2D}$ implies that (see [Horn and Johnson 1985]):

(a) The eigenvalues of $\Gamma_{2D}$ are equal to the products of the eigenvalues of $\Gamma_{1D}$ and $\Gamma_{1D}$.

(b) The eigenvectors of $\Gamma_{2D}$ are given by the Kronecker products of the eigenvectors of $\Gamma_{1D}$ and $\Gamma_{1D}$.

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, and the references cited there, for details on 1D Slepian taper computation.

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

$$\tilde{y}(t) = [y(t), y(t-1), \ldots, y(t-m)]^T$$

and $a(\omega)$ is the 1D frequency response. Specifically, consider the LS problem:

$$\min_{\beta} \sum_{t=m+1}^{N} [\tilde{y}(t) - a(\omega) e^{i\omega t}]^* W^{-1} [\tilde{y}(t) - a(\omega) e^{i\omega t}]$$

where $W$ is a positive definite weight matrix.
where $W^{-1}$ is a weighting matrix which is yet to be specified, and where

$$a(\omega) = \begin{bmatrix} 1, e^{-i\omega}, \ldots, e^{-im\omega} \end{bmatrix}^T$$

(5.6.82)

Note that the definition of $a(\omega)$ in (5.6.82) differs from that of $a(\omega)$ in (5.6.74). The solution to (5.6.81) is given by

$$\beta(\omega) = \frac{a^*(\omega)W^{-1}g(\omega)}{a^*(\omega)W^{-1}a(\omega)}$$

(5.6.83)

where

$$g(\omega) = \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t)e^{-i\omega t}$$

(5.6.84)

For

$$W = \hat{R} \triangleq \frac{1}{N-m} \sum_{t=m+1}^{N} \tilde{y}(t)\tilde{y}^*(t)$$

(5.6.85)

the weighted LS estimate of the amplitude spectrum in (5.6.83) reduces to the Capon method (see equation (5.6.55) in Complement 5.6.4), whereas for

$$W = \hat{R} - g(\omega)g^*(\omega) \triangleq \hat{Q}(\omega)$$

(5.6.86)

equation (5.6.83) gives the APES method (see equations (5.6.48), (5.6.49), and (5.6.51) 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 (5.6.80), let

$$[y(t-k, \tilde{t}-\tilde{k})] = \begin{bmatrix} y(t, \tilde{t}) & \cdots & y(t, \tilde{t} - \tilde{m}) \\ \vdots & & \vdots \\ y(t-m, \tilde{t}) & \cdots & y(t-m, \tilde{t} - \tilde{m}) \end{bmatrix}$$

(5.6.87)

be the 2D data matrix, and let

$$\tilde{y}(t, \tilde{t}) = \text{vec} \left( [y(t-k, \tilde{t}-\tilde{k})] \right)$$

$$= [y(t, \tilde{t}), \ldots, y(t-m, \tilde{t})] \ldots [y(t, \tilde{t} - \tilde{m}), \ldots, y(t-m, \tilde{t} - \tilde{m})]^T$$

(5.6.88)

Our goal is to fit the data matrix in (5.6.87) to the matrix corresponding to a generic 2D sinusoid with frequency pair $(\omega, \tilde{\omega})$, that is:

$$\begin{bmatrix} e^{i[\omega(t-k)+\tilde{\omega}(\tilde{t}-\tilde{k})]} & \cdots & e^{i[\omega(t-k)+\tilde{\omega}(\tilde{t} - \tilde{m})]} \\ \vdots & & \vdots \\ e^{i[\omega(t-m)+\tilde{\omega}(\tilde{t})]} & \cdots & e^{i[\omega(t-m)+\tilde{\omega}(\tilde{t} - \tilde{m})]} \end{bmatrix}$$

(5.6.89)

Similarly to (5.6.88), let us vectorize (5.6.89):

$$\text{vec} \left( [e^{i[\omega(t-k)+\tilde{\omega}(\tilde{t}-\tilde{k})]}] \right) = \beta e^{i(\omega t + \tilde{\omega} \tilde{t})} \text{vec} \left( [e^{-i(\omega k + \tilde{\omega} \tilde{k})}] \right)$$

$$= \beta e^{i(\omega t + \tilde{\omega} \tilde{t})} \tilde{a}(\tilde{\omega}) \otimes a(\omega)$$

(5.6.90)
As in (5.6.76), let
\[ b(\omega, \tilde{\omega}) = \tilde{a}(\omega) \otimes a(\omega), \quad (m + 1)(\bar{m} + 1) \times 1 \quad (5.6.91) \]

We deduce from (5.6.88)–(5.6.91) that the 2D counterpart of the 1D weighted LS fitting problem in (5.6.81) is the following:
\[
\min_{\beta} \sum_{t=m+1}^{N} \sum_{\tilde{t}=\bar{m}+1}^{\bar{N}} \left[ \tilde{y}(t, \tilde{t}) - \beta e^{i(\omega t + \tilde{\omega} \tilde{t})} b(\omega, \tilde{\omega}) \right]^* W^{-1} \cdot \left[ \tilde{y}(t, \tilde{t}) - \beta e^{i(\omega t + \tilde{\omega} \tilde{t})} b(\omega, \tilde{\omega}) \right] \quad (5.6.92)
\]

The solution to (5.6.92) is given by:
\[
\beta(\omega, \tilde{\omega}) = \frac{b^*(\omega, \tilde{\omega}) W^{-1} g(\omega, \tilde{\omega})}{b^*(\omega, \tilde{\omega}) W^{-1} b(\omega, \tilde{\omega})} \quad (5.6.93)
\]

where
\[
g(\omega, \tilde{\omega}) = \frac{1}{(N-m)(\bar{N}-\bar{m})} \sum_{t=m+1}^{N} \sum_{\tilde{t}=\bar{m}+1}^{\bar{N}} \tilde{y}(t, \tilde{t}) e^{-i(\omega t + \tilde{\omega} \tilde{t})} \quad (5.6.94)
\]

The 2D Capon method is given by (5.6.93) with
\[
W = \frac{1}{(N-m)(\bar{N}-\bar{m})} \sum_{t=m+1}^{N} \sum_{\tilde{t}=\bar{m}+1}^{\bar{N}} \tilde{y}(t, \tilde{t}) \tilde{y}^*(t, \tilde{t}) \triangleq \hat{R} \quad (5.6.95)
\]

whereas the 2D APES method is given by (5.6.93) with
\[
W = \hat{R} - g(\omega, \tilde{\omega}) g^*(\omega, \tilde{\omega}) \triangleq \hat{Q}(\omega, \tilde{\omega}) \quad (5.6.96)
\]

Note that \( g(\omega, \tilde{\omega}) \) in (5.6.94) can be efficiently evaluated using a 2D FFT algorithm. However, an efficient implementation of the 2D spectral estimate in (5.6.93) is not so direct. A naive implementation may be rather time consuming owing to the large dimensions of the vectors and matrices involved, as well as the need to evaluate \( \beta(\omega, \tilde{\omega}) \) 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.
To proceed, some additional notation is required. Let

\[ g_n(z) = 1 + \alpha_1 z^{-1} + \cdots + \alpha_{m-n-1} z^{-(m-n)} \]

and let \( y(t) \) be a signal whose PSD is equal to \( |g_s(\omega)|^2 \); hence, \( y(t) \) is an \( n \)th-order MA process. By making use of (1.3.9) and (1.4.9), along with the above 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 \( \omega \) 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 which 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 \( \theta \). The result is obtained from equation (6.4.7), which for \( n = 1 \) reduces to:

\[
\hat{\theta} = \arg \max_{\theta} a^*(\theta) \hat{R} a(\theta) = \arg \max_{\theta} \sum_{t=1}^{N} |a^*(\theta)y(t)|^2 \tag{6.5.9}
\]
Section 6.5  Complements  289

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:

\[
s(t) = \alpha e^{i\phi(t)} \tag{6.5.10}
\]

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 which is to be minimized with respect to \( \{\phi(t)\}_t^{N}, \alpha, \) and \( \theta \):

\[
\sum_{t=1}^{N} \left\| y(t) - \alpha e^{i\phi(t)} a(\theta) \right\|^2 \\
= \sum_{t=1}^{N} \left\{ \|y(t)\|^2 + \alpha^2 \|a(\theta)\|^2 - 2\alpha \text{Re} \left[ a^*(\theta)y(t)e^{-i\phi(t)} \right] \right\} \tag{6.5.11}
\]

It follows from (6.5.11) that the NLS estimate of \( \{\phi(t)\}_t^{N} \) is given by the maximizer of the function:

\[
\text{Re} \left[ a^*(\theta)y(t)e^{-i\phi(t)} \right] = \text{Re} \left[ |a^*(\theta)y(t)| e^{i\text{arg}[a^*(\theta)y(t)]} e^{-i\phi(t)} \right] \\
= |a^*(\theta)y(t)| \cos \left[ \text{arg}(a^*(\theta)y(t)) - \phi(t) \right] \tag{6.5.12}
\]

which is easily seen to be

\[
\hat{\phi}(t) = \text{arg} \left[ a^*(\theta)y(t) \right], \quad t = 1, \ldots, N \tag{6.5.13}
\]

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 \( \theta \) for the constant modulus signal case is given by:

\[
\hat{\theta} = \arg \max_{\theta} \sum_{t=1}^{N} |a^*(\theta)y(t)| \tag{6.5.14}
\]

Finally, the NLS estimate of \( \alpha \) is obtained by minimizing (6.5.11) (with \( \{\phi(t)\} \) and \( \theta \) replaced by (6.5.13) and (6.5.14), respectively):

\[
\hat{\alpha} = \frac{1}{\sum_{t=1}^{N} \left| a^*(\hat{\theta})y(t) \right|} \sum_{t=1}^{N} \left| a^*(\hat{\theta})y(t) \right| \tag{6.5.15}
\]
Remark: It follows easily from the above derivation that if \( \alpha \) is known (which may be the case when the emitted signal has a known amplitude that is not significantly distorted during propagation), the NLS estimates of \( \theta \) 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 data-independent. In contrast, the Capon spatial filter is data-dependent, or data-adaptive; see equation (6.3.24). It is this data-adaptivity that confers to the Capon method better resolution and significantly reduced leakage 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, which are 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 \( \theta \) denote a generic DOA, and consider equation (6.2.19):

\[
y(t) = a(\theta)s(t) + e(t)
\]  

that describes the array output, \( y(t) \), as a sum of a possible signal component impinging from the generic DOA \( \theta \) and a term \( e(t) \) that includes noise and any other signals with DOAs different from \( \theta \). Let \( \sigma_s^2 \) denote the power of the signal \( s(t) \) in (6.5.16), which is the main parameter we want to estimate: \( \sigma_s^2 \) as a function of \( \theta \) provides an estimate of the spatial spectrum. Let us estimate the spatial filter vector, \( h \), as well as the signal power, \( \sigma_s^2 \), by solving the following least squares (LS) problem:

\[
\min_{h, \sigma_s^2} E \left\{ |h^* y(t) - s(t)|^2 \right\}
\]  

(6.5.17)

Of course, the signal \( s(t) \) in (6.5.17) is not known. However, as we show below, (6.5.17) does not depend on \( s(t) \) but only on its power \( \sigma_s^2 \), 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:

\[
E \{ y(t) s^*(t) \} = a(\theta) \sigma_s^2
\]

which implies that

\[
E \{ |h^* y(t) - s(t)|^2 \} = h^* R h - h^* a(\theta) \sigma_s^2 - a^*(\theta) h \sigma_s^2 + \sigma_s^2
\]

\[
= [h - \sigma_s^2 R^{-1} a(\theta)]^* R [h - \sigma_s^2 R^{-1} a(\theta)] + \sigma_s^2 [1 - \sigma_s^2 a^*(\theta) R^{-1} a(\theta)]
\]

(6.5.19)

Omitting the trivial solution \( h = 0, \sigma_s^2 = 0 \), the minimization of (6.5.19) with respect to \( h \) and \( \sigma_s^2 \) yields:

\[
h = \frac{R^{-1} a(\theta)}{a^*(\theta) R^{-1} a(\theta)}
\]

(6.5.20)

\[
\sigma_s^2 = \frac{1}{a^*(\theta) R^{-1} a(\theta)}
\]

(6.5.21)

which coincides with the Capon solution in (6.3.24) and (6.3.25). To obtain \( \sigma_s^2 \) 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 \( \sigma_s^2 \).

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 \( f_s(t) \) 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

\[
\sigma_s^2 a(\theta) a^*(\theta)
\]

(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:

\[
\min_{\sigma_s^2} \| R - \sigma_s^2 a(\theta) a^*(\theta) \|^2
\]

\[
= \min_{\sigma_s^2} \{ \text{constant} + \sigma_s^2 [a^*(\theta) a(\theta)]^2 - 2 \sigma_s^2 a^*(\theta) R a(\theta) \}
\]

(6.5.23)

As \( a^*(\theta) a(\theta) = m \) (by assumption; see (6.3.11)), it follows from (6.5.23) that the minimizing \( \sigma_s^2 \) is given by:

\[
\sigma_s^2 = \frac{1}{m^2} a^*(\theta) R a(\theta)
\]

(6.5.24)

which coincides with the Beamforming estimate of the power coming from DOA \( \theta \) (see (6.3.16)).
Chapter 6 Spatial Methods

To obtain the Capon method by following a similar idea to the one above, we fit the pseudoinverse of (6.5.22) to the inverse of $R$:

$$\min_{\sigma_s^2} \left\| R^{-1} - \left[ \sigma_s^2 a(\theta)a^*(\theta) \right]^\dagger \right\|^2$$  \hspace{1cm} (6.5.25)

It is easily verified that the Moore-Penrose pseudoinverse of $\sigma_s^2 a(\theta)a^*(\theta)$ is given by

$$\left[ \sigma_s^2 a(\theta)a^*(\theta) \right]^\dagger = \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{a^*(\theta)a(\theta)} = \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{m^2}$$  \hspace{1cm} (6.5.26)

This follows, for instance, from (A.8.8) and the fact that

$$\sigma_s^2 a(\theta)a^*(\theta) = \left[ \sigma_s^2 \| a(\theta) \|^2 \right] \left[ \frac{a(\theta)}{\| a(\theta) \|} \right] \left[ \frac{a(\theta)}{\| a(\theta) \|} \right]^* \triangleq \sigma_{uv}^*$$  \hspace{1cm} (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

$$\min_{\sigma_s^2} \left\| R^{-1} - \frac{1}{\sigma_s^2} \frac{a(\theta)a^*(\theta)}{m^2} \right\|^2$$  \hspace{1cm} (6.5.28)

whose solution, by analogy with (6.5.23)–(6.5.24), is given by the Capon estimate of the signal power:

$$\sigma_s^2 = \frac{1}{a^*(\theta)R^{-1}a(\theta)}$$  \hspace{1cm} (6.5.29)

It is worth noting that in the present covariance fitting-based derivation, the signal power $\sigma_s^2$ 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:

$$\min_{\sigma_s^2} \left\| R^{-1/2} \left[ R - \sigma_s^2 a(\theta)a^*(\theta) \right] R^{-1/2} \right\|^2$$  \hspace{1cm} (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:

$$\left\| I - \sigma_s^2 R^{-1/2}a(\theta)a^*(\theta)R^{-1/2} \right\|^2 = \text{constant} - 2\sigma_s^4 \sigma_{uv}^* R^{-1}a(\theta) + \sigma_s^4 \left[a^*(\theta)R^{-1}a(\theta)\right]^2$$  \hspace{1cm} (6.5.31)
The minimization of (6.5.31) with respect to $\sigma_s^2$ 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 which is 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 $\sigma_s^2$ for which this condition holds. Mathematically, $\sigma_s^2$ is the solution to the following constrained covariance fitting problem:

$$\max \sigma_s^2 \quad \text{subject to} \quad R - \sigma_s^2 a(\theta) a^*(\theta) \succeq 0$$  \hspace{1cm} (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:

$$R - \sigma_s^2 a(\theta) a^*(\theta) \succeq 0$$
$$\iff I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2} \succeq 0$$
$$\iff 1 - \sigma_s^2 a^*(\theta) R^{-1} a(\theta) \geq 0$$
$$\iff \sigma_s^2 \leq \frac{1}{a^*(\theta) R^{-1} a(\theta)}$$  \hspace{1cm} (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 one 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

$$h^* \left[ R - \sigma_s^2 a(\theta) a^*(\theta) \right] h \geq 0 \quad \text{for any} \quad h \in \mathbb{C}^{m \times 1}$$  \hspace{1cm} (6.5.34)

which, in turn, is equivalent to

$$h^* \left[ R - \sigma_s^2 a(\theta) a^*(\theta) \right] h \geq 0$$
$$\quad \text{for any} \quad h \quad \text{such that} \quad h^* a(\theta) = 1$$  \hspace{1cm} (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 \( h/\alpha^* \) 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

\[
h^*[R - \sigma_s^2 a(\theta)a^*(\theta)] h = h^*Rh \geq 0
\]

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

\[
h^*Rh \geq \sigma_s^2 \quad \text{for any } h \text{ such that } h^*a(\theta) = 1 \quad (6.5.36)
\]

From (6.5.36) we can see that the solution to (6.5.32) is given by

\[
\sigma_s^2 = \min_h h^*Rh \quad \text{subject to } h^*a(\theta) = 1
\]

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 imprecisely known.

### 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 accurately known. However, whenever the knowledge of \( a(\theta) \) is imprecise, the performance of the Capon method may become worse than that of the Beamforming method. To see why this is so, consider a scenario in which the problem is to determine the power coming from a source with DOA assumed to be equal to \( \theta_0 \). Let us assume that in actuality the true DOA of the source is \( \theta_0 + \Delta \). For the Capon beamformer pointed toward \( \theta_0 \), 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 \( \Delta \) 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 following uncertainty ellipsoid:

\[
(a - \bar{a})^*C^{-1}(a - \bar{a}) \leq 1 \quad (6.5.37)
\]

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 \( \theta \); however, for the sake of notational convenience, we drop the \( \theta \) dependence of these variables.
In some applications there may 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

$$||a - \bar{a}||^2 \leq \varepsilon$$

(6.5.38)

where $\varepsilon$ 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 that the performance of the robust Capon method remains almost unchanged when $\varepsilon$ is varied in a relatively large interval around $\varepsilon_0$ (see [Stoica, Wang, and Li 2003], [Li, Stoica, and Wang 2003]).

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 $\sigma_s^2$ and $a$ by solving the following constrained covariance fitting problem:

$$\max_{\sigma_s^2, a} \sigma_s^2 \text{ subject to: } R - \sigma_s^2 aa^* \geq 0$$

$$a - \bar{a} \text{ subject to: } a - \bar{a} C^{-1} (a - \bar{a}) \leq 1$$

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

$$\bar{a}^* C^{-1} \bar{a} > 1$$

(6.5.40)

(which is a regularity condition).

Because both $\sigma_s^2$ and $a$ are considered to be free parameters in the above fitting problem, there is a scaling ambiguity in the signal covariance term in (6.5.39), in the sense that both $(\sigma_s^2, a)$ 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 condition (see (6.3.11)):

$$a^* a = m$$

(6.5.41)

However, the constraint in (6.5.41) is non-convex, which 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 quite efficiently solved, as we show below. 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 $a$ which satisfies (6.5.41) by scaling $\tilde{a}$:

$$\bar{a} = \frac{\sqrt{m}}{||\tilde{a}||} \tilde{a}$$

and a corresponding estimate of $\sigma^2$ by scaling $\tilde{\sigma}_s^2$ such that the signal covariance term is left unchanged, i.e., $\tilde{\sigma}_s^2 \bar{a} \bar{a}^* = \sigma^2 \bar{a} \bar{a}^*$, which gives:
To derive the solution \( (\tilde{\sigma}_s^2, \tilde{a}) \) of (6.5.39) we first note that, for any fixed \( a \), the maximizing \( \sigma_s^2 \) is given by

\[
\tilde{\sigma}_s^2 = \frac{1}{a^* R^{-1} a}
\]

(see equation (6.5.33) in Complement 6.5.3). This simple observation allows us to eliminate \( \sigma_s^2 \) from (6.5.39) and hence reduce (6.5.39) to the following problem:

\[
\min_a a^* R^{-1} a \quad \text{subject to:} \quad (a - \tilde{a})^* C^{-1} (a - \tilde{a}) \leq 1
\]

(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, and therefore we can reformulate (6.5.44) as the following quadratic problem with a quadratic equality constraint

\[
\min_a a^* R^{-1} a \quad \text{subject to:} \quad (a - \tilde{a})^* C^{-1} (a - \tilde{a}) = 1
\]

(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):

\[
\min_a a^* R^{-1} a \quad \text{subject to:} \quad \|a - \tilde{a}\|^2 = \varepsilon
\]

(6.5.46)

(the case of \( C \neq \varepsilon I \) can be similarly treated). Define

\[
x = a - \tilde{a}
\]

and rewrite (6.5.46) using \( x \) in lieu of \( a \):

\[
\min_x [x^* R^{-1} x + x^* R^{-1} \tilde{a} + \tilde{a}^* R^{-1} x] \quad \text{subject to:} \quad \|x\|^2 = \varepsilon
\]

(6.5.48)

Owing to the constraint in (6.5.48), the \( x \) that solves (6.5.48) is also a solution to the problem:

\[
\min_x [x^* (R^{-1} + \lambda I) x + x^* R^{-1} \tilde{a} + \tilde{a}^* R^{-1} x] \quad \text{subject to:} \quad \|x\|^2 = \varepsilon
\]

(6.5.49)

where \( \lambda \) is an arbitrary constant. Let us consider a particular choice of \( \lambda \), which is a solution of the equation:

\[
\tilde{a}^* (I + \lambda R)^{-2} \tilde{a} = \varepsilon
\]

(6.5.50)

and which is also such that

\[
R^{-1} + \lambda I > 0
\]

(6.5.51)

Then, the unconstrained minimizer of the function in (6.5.49) is given by

\[
x = -(R^{-1} + \lambda I)^{-1} R^{-1} \tilde{a} = -(I + \lambda R)^{-1} \tilde{a}
\]

(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 $\lambda$ 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, which we do next.

Let

$$R = UAU^*$$  \hfill (6.5.53)

denote the eigenvalue decomposition (EVD) of $R$, where $U^*U = UU^* = I$ and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_m \end{bmatrix}; \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$$  \hfill (6.5.54)

Also, let

$$b = U^* \bar{a}$$  \hfill (6.5.55)

Using (6.5.53)–(6.5.55) we can rewrite the left-hand side of equation (6.5.50) as:

$$g(\lambda) \triangleq \bar{a}^* [I + \lambda R]^{-2} \bar{a} = \bar{a}^* [U(I + \lambda \Lambda)U^*]^{-2} \bar{a}$$

$$= b^* (I + \lambda \Lambda)^{-2} b = \sum_{k=1}^{m} \frac{|b_k|^2}{(1 + \lambda \lambda_k)^2}$$  \hfill (6.5.56)

where $b_k$ is the $k$th element of the vector $b$. Note that

$$\sum_{k=1}^{m} |b_k|^2 = ||b||^2 = ||\bar{a}||^2 > \varepsilon$$  \hfill (6.5.57)

(see (6.5.55) and (6.5.40)). It follows from (6.5.56) and (6.5.57) that $\lambda$ can be a solution of the equation $g(\lambda) = \varepsilon$ only if

$$(1 + \lambda \lambda_k)^2 > 1$$  \hfill (6.5.58)

for some value of $k$. At the same time, $\lambda$ should be such that (see (6.5.51)):

$$R^{-1} + \lambda I > 0 \iff I + \lambda R > 0$$

$$\iff 1 + \lambda \lambda_k > 0 \text{ for } k = 1, \ldots, m$$  \hfill (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

$$\lambda > 0$$  \hfill (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 \textit{monotonically decreasing function} of $\lambda$ for $\lambda > 0$. Furthermore, for

$$\lambda_L = \frac{||\bar{a}|| - \sqrt{\varepsilon}}{\lambda_1 \sqrt{\varepsilon}}$$  \hfill (6.5.61)

we have that

$$g(\lambda_L) > \frac{1}{(1 + \lambda_L \lambda_1)^2} ||b||^2 = \frac{\varepsilon}{||\bar{a}||^2} = \varepsilon$$  \hfill (6.5.62)
Similarly, for
\[ \lambda_U = \frac{\|\bar{a}\| - \sqrt{\varepsilon}}{\lambda_m \sqrt{\varepsilon}} \geq \lambda_L \]  
we can verify that
\[ g(\lambda_U) < \frac{1}{(1 + \lambda_U \lambda_m)^2} \|b\|^2 = \varepsilon \]  
Summarizing the previous facts, it follows that equation (6.5.50) has a unique solution for \( \lambda \) 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.

**Robust Capon Algorithm**

**Step 1.** Compute the eigendecomposition \( \mathbf{R} = \mathbf{U} \Lambda \mathbf{U}^* \) and set \( \mathbf{b} = \mathbf{U}^* \bar{a} \).
**Step 2.** Solve the equation \( g(\lambda) = \varepsilon \) for \( \lambda \) 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)):
\[ \tilde{a} = \bar{a} - \mathbf{U}(I + \lambda \Lambda)^{-1} \mathbf{b} \]  
and, finally, compute the power estimate (see (6.5.42) and (6.5.43))
\[ \hat{\sigma}_s^2 = \frac{\tilde{a}^* \bar{a}}{m \tilde{a}^* \mathbf{U} \Lambda^{-1} \mathbf{U}^* \bar{a}} \]  
where, from (6.5.65), \( \mathbf{U}^* \bar{a} = \mathbf{b} - (I + \lambda \Lambda)^{-1} \mathbf{b} \).

The bulk of the computation in the algorithm involves computing the EVD of \( \mathbf{R} \), which requires \( \mathcal{O}(m^3) \) arithmetic operations. Hence, the computational complexity of the above 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, as well as many numerical examples illustrating the good performance of the Robust Capon method, including its insensitivity to the choice of \( \varepsilon \) in (6.5.38) or \( C \) in (6.5.37).

### 6.5.5 Capon Method with Noise Gain Constraint

As explained in Complement 6.5.4, the Capon method performs poorly as a power estimator in the presence of steering vector errors (yet, it may 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 small-sample errors, see e.g. [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 imprecisely known. 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 following matrix:

$$R + \lambda I \tag{6.5.67}$$

where the diagonal loading factor $\lambda > 0$ is a user-selected parameter. The so-obtained filter vector $h$ 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 to either a theoretical covariance matrix or a sample covariance matrix.

There have been several rules proposed in the literature for choosing the parameter $\lambda$ in (6.5.68). Most of these rules choose $\lambda$ 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 $\lambda$ is desired.

One commonly-used data-dependent rule selects the diagonal loading factor $\lambda > 0$ that satisfies

$$||h||^2 = \frac{a^*(R + \lambda I)^{-2}a}{a^*(R + \lambda I)^{-1}a}^2 = c \tag{6.5.69}$$

where the constant $c$ must be chosen by the user. Let us explain briefly why choosing $\lambda$ 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) may run out of “degrees of freedom” and hence may not pay enough attention to the noise in the data (unless the SNR is very
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 \( c \). While the choice of \( c \) may be easier than the direct choice of \( \lambda \) in (6.5.68), it is far from trivial, and in fact clear-cut rules for selecting \( c \) are hardly available. In particular, a “too small” value of \( c \) may 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:

\[
\min_h h^*Rh \quad \text{subject to: } h^*a = 1 \text{ and } \|h\|^2 \leq c
\]  

(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). While 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]), we present a rigorous and thorough analysis of this solution. As a byproduct, the following analysis 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 \( \theta \); to simplify notation we will omit the functional dependence on \( \theta \) 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:

\[
a = \hat{a} - (I + \lambda R)^{-1}\hat{a} = (I + \lambda R)^{-1}[(I + \lambda R) - I] \hat{a}
\]

\[
= (\frac{1}{\lambda} R^{-1} + I)^{-1} \hat{a}
\]

(6.5.71)

The RCM estimates the signal power by

\[
\frac{1}{a^*R^{-1}a}
\]

(6.5.72)

with \( a \) as given in (6.5.71) above, and hence RCM does not directly use any spatial filter. However, the power estimate in (6.5.72) is equal to \( h^*Rh \), where

\[
h = \frac{R^{-1}a}{a^*R^{-1}a}
\]

(6.5.73)

and 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:

\[
h = \frac{(R + \frac{1}{\lambda} I)^{-1} a}{a^* [(R + \frac{1}{\lambda} I) R^{-1} (R + \frac{1}{\lambda} I)]^{-1} a}
\]

(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 \( \lambda \) 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 which 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 $\varepsilon$ 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 may 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 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):

$$S = \{ h \mid h^* a = 1 \text{ and } \|h\|^2 \leq c \} \quad (6.5.75)$$

By the Cauchy–Schwarz inequality (see Result R12 in Appendix A), we have that:

$$1 = |h^* a|^2 \leq \|h\|^2 \|a\|^2 \leq cm \implies c \geq \frac{1}{m} \quad (6.5.76)$$

where we also used the fact that (by assumption; see (6.3.11))

$$\|a\|^2 = m \quad (6.5.77)$$

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 following norm:

$$\|h_{CM}\|^2 = \frac{a^* R^{-2} a}{(a^* R^{-1} a)^2} \quad (6.5.78)$$

As the noise gain of the CM is typically too high, we should like to choose $c$ so that

$$c < \frac{a^* R^{-2} a}{(a^* R^{-1} a)^2} \quad (6.5.79)$$

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 it is the solution to the CCM problem. Combining
(6.5.76) and (6.5.79) yields the following interval for $c$:

$$c \in \left[ \frac{1}{m}, \frac{a^* R^{-2} a}{(a^* R^{-1} a)^2} \right]$$

(6.5.80)

Similarly to (6.5.53), let

$$R = U \Lambda U^*$$

(6.5.81)

be the eigenvalue decomposition (EVD) of $R$, where $U^* U = U U^* = I$ and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ \vdots & \ddots \\ 0 & \lambda_m \end{bmatrix}; \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$$

(6.5.82)

As

$$\frac{a^* R^{-2} a}{[a^* R^{-1} a]^2} \leq \frac{\|a\|^2 / \lambda_m^2}{\|a\|^2 / \lambda_1^2} = \frac{\lambda_1^2}{m \lambda_m^2}$$

(6.5.83)

it follows from (6.5.79) that $c$ also satisfies:

$$mc < \frac{\lambda_1^2}{\lambda_m^2}$$

(6.5.84)

The above inequality will be useful later on.

Next, let us define the function

$$g(h, \lambda, \mu) = h^* R h + \lambda(\|h\|^2 - c) + \mu(-h^* a - a^* h + 2)$$

(6.5.85)

where $\mu \in \mathbb{R}$ is arbitrary and where

$$\lambda > 0$$

(6.5.86)

**Remark:** We note in passing that $\lambda$ and $\mu$ are the so-called Lagrange multipliers, and $g(h, \lambda, \mu)$ 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:

$$g(h, \lambda, \mu) \leq h^* R h \quad \text{for any } h \in S$$

(6.5.87)

and for any $\mu \in \mathbb{R}$ and $\lambda > 0$. The part of (6.5.85) that depends on $h$ can be written as

$$h^* (R + \lambda I) h - \mu h^* a - \mu a^* h$$

$$= \left[ h - \mu (R + \lambda I)^{-1} a \right]^* (R + \lambda I) \left[ h - \mu (R + \lambda I)^{-1} a \right]$$

$$- \mu^2 a^* (R + \lambda I)^{-1} a$$

(6.5.88)
Section 6.5 Complements

Hence, for fixed $\lambda$ and $\mu$, the *unconstrained minimizer* of $g(h, \lambda, \mu)$ with respect to $h$ is given by:

$$\hat{h}(\lambda, \mu) = \mu(R + \lambda I)^{-1}a$$  (6.5.89)

Let us choose $\mu$ such that (6.5.89) satisfies the first constraint in (6.5.70):

$$\hat{h}^*(\lambda, \mu)a = 1 \iff \hat{\mu} = \frac{1}{a^*(R + \lambda I)^{-1}a}$$  (6.5.90)

(which is always possible, for $\lambda > 0$). Also, let us choose $\lambda$ so that (6.5.89) also satisfies the second constraint in (6.5.70) *with equality*, i.e.,

$$\|\hat{h}(\hat{\lambda}, \hat{\mu})\|^2 = c \iff \frac{a^*(R + \hat{\lambda}I)^{-2}a}{a^*(R + \hat{\lambda}I)^{-1}a} = c$$  (6.5.91)

We will show shortly that the above 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)):

$$\hat{h}(\hat{\lambda}, \hat{\mu}) = (R + \hat{\lambda}I)^{-1}a$$  (6.5.92)

As $\hat{\lambda}$ satisfies (6.5.91), the above vector $\hat{h}(\hat{\lambda}, \hat{\mu})$ lies on the boundary of $S$, and hence (see also (6.5.87)):

$$g\left(\hat{h}(\hat{\lambda}, \hat{\mu}), \hat{\lambda}, \hat{\mu}\right) = \hat{h}^*(\hat{\lambda}, \hat{\mu})Rh(\hat{\lambda}, \hat{\mu}) \leq h^*Rh \quad \text{for any } h \in S$$  (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:

$$f(\lambda) = c$$  (6.5.94)

where

$$f(\lambda) = \frac{a^*(R + \lambda I)^{-2}a}{\left[a^*(R + \lambda I)^{-1}a\right]^2} = \frac{\sum_{k=1}^{m} |b_k|^2}{\sum_{k=1}^{m} \frac{|\lambda_k + \lambda|^2}{|\lambda_k + \lambda|}^2}$$  (6.5.95)

and where $b_k$ is the $k$th element of the vector

$$b = U^*a$$  (6.5.96)
Differentiation of (6.5.95) with respect to $\lambda$ yields:

$$
f' (\lambda) = \left\{ -2 \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^3} \right] \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right]^2 
+ 2 \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^2} \right] \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right] \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^2} \right] \right\} 
\cdot \frac{1}{\left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right]^4} 
= -2 \left\{ \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^2} \right] \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right] - \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^2} \right]^2 \right\} 
\cdot \frac{1}{\left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right]^4} 
$$

(6.5.97)

Making use of the Cauchy–Schwartz inequality once again, we can show that

$$
\left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^2} \right] = \left[ \sum_{k=1}^{m} \frac{|b_k|}{(\lambda_k + \lambda)^{3/2}} \frac{|b_k|}{(\lambda_k + \lambda)^{1/2}} \right]^2 
\leq \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)^3} \right] \left[ \sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k + \lambda)} \right] 
$$

(6.5.98)

Hence,

$$
f' (\lambda) < 0 \text{ for any } \lambda > 0 
\text{ (and } \lambda_k \neq \lambda_p \text{ for at least one pair } k \neq p) 
$$

(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 $\lambda$ would also be useful. Such a bound can be obtained as follows. A simple calculation shows that

$$
c = f(\lambda) < \frac{||b||^2}{(\lambda_m + \lambda)^2} = \frac{(\lambda_1 + \lambda)^2}{m(\lambda_m + \lambda)^2} 
\implies mc(\lambda_m + \lambda)^2 < (\lambda_1 + \lambda)^2 
$$

(6.5.100)
where we used the fact that $\|b\|^2 = \|a\|^2 = m$. From (6.5.100) we see that $\lambda$ must satisfy the inequality

$$\lambda < \frac{\lambda_1 - \sqrt{mc}\lambda_m}{\sqrt{mc} - 1} \triangleq \lambda_U \quad (6.5.101)$$

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.

### Constrained Capon Algorithm

**Step 1.** Compute the eigendecomposition $R = U\Lambda U^* \text{ and set } b = U^*a$.

**Step 2.** Solve the equation $f(\lambda) = c$ for $\lambda$ using, e.g., a Newton method along with the fact that there is a unique solution which lies in the interval $(0, \lambda_U)$.

**Step 3.** Compute the (diagonally-loaded) spatial filter vector

$$h = \frac{\lambda + \Lambda}{\lambda + \Lambda}^{-1}a = \frac{U(R + \Lambda I)^{-1}b}{b^*(R + \Lambda I)^{-1}b}$$

where $\lambda$ is found in Step 2, and estimate the signal power as $h^*Rh$.

To conclude this complement, we note that the above CCM algorithm is quite similar to the RCM algorithm presented in Complement 6.5.4. The only differences are that the equation for $\lambda$ associated with the CCM is slightly more complicated, and more importantly, that it is harder to select $c$ needed in the CCM (for any DOA of interest) than it is to select $\varepsilon$ 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 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 $\theta$ in a distortionless manner, and at the same time attenuates signals with DOAs different from $\theta$ 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 $\theta$ denote a generic DOA and consider the equation (6.2.19),

$$y(t) = a(\theta)s(t) + e(t), \quad t = 1, \ldots, N \quad (6.5.102)$$

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 $\theta$, and a term, $e(t)$, that includes noise along with any other signals whose DOAs are different from $\theta$. We assume that the
array is \textit{uniform and linear}, in which case $a(\theta)$ is given by

$$a(\theta) = \left[1, e^{-i\omega_s}, \ldots, e^{-i(m-1)\omega_s}\right]^T \quad (6.5.103)$$

where $m$ denotes the number of sensors in the array, and $\omega_s = (\omega_d \sin \theta) / c$ 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. While this is a limitation, it is not a serious one because there are techniques which can be used to approximately transform the direction vector of a general array into the direction vector of a fictitious ULA (see, e.g., [Doron, Doron, and Weiss 1993]). Such a technique performs a relatively simple DOA-independent linear transformation of the array output snapshots; the so-obtained 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 $\sigma_s^2$ 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 $\theta$, provides an estimate of the spatial spectrum. In this complement, we assume that $\{s(t)\}_{t=1}^N$ is an unknown \textit{deterministic} sequence, and hence we define $\sigma_s^2$ as

$$\sigma_s^2 = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^N |s(t)|^2 \quad (6.5.104)$$

An important difference between equation (6.5.102) and its temporal counterpart (see, e.g., equation (5.6.81) in Complement 5.6.6) is that in (6.5.102) the signal $s(t)$ is \textit{completely unknown}, whereas in the temporal case we had $s(t) = be^{i\omega t}$ 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.

\textbf{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:

$$\bar{a}_k = \left[e^{-i(k-1)\omega_s}, e^{-ik\omega_s}, \ldots, e^{-i(k+\bar{m}-2)\omega_s}\right]^T \quad (\bar{m} \times 1) \quad (6.5.105)$$

$$\bar{y}_k(t) = [y_k(t), y_{k+1}(t), \ldots, y_{k+\bar{m}-1}(t)]^T \quad (\bar{m} \times 1) \quad (6.5.106)$$

for $k = 1, \ldots, L$, with

$$L = m - \bar{m} + 1 \quad (6.5.107)$$

In (6.5.106), $y_k(t)$ denotes the $k$th element of $y(t)$; also, we omit the dependence of $\bar{a}_k$ on $\theta$ to simplify notation. The choice of the user parameter $\bar{m}$ will be discussed later.
Section 6.5 Complements

Owing to the assumed ULA structure, the direction subvectors \{\tilde{a}_k\} satisfy the following relations:

\[ \tilde{a}_k = e^{-i(k-1)\omega} \tilde{a}_1, \quad k = 2, \ldots, L \]  

(6.5.108)

Consequently, \( \tilde{y}_k(t) \) can be written as (see (6.5.102)):

\[ \tilde{y}_k(t) = \tilde{a}_k s(t) + \tilde{e}_k(t) = e^{-i(k-1)\omega} \tilde{a}_1 s(t) + \tilde{e}_k(t) \]  

(6.5.109)

where \( \tilde{e}_k(t) \) is a noise vector defined similarly to \( \tilde{y}_k(t) \). Let \( h \) denote the \((\tilde{m} \times 1)\) coefficient vector of a spatial filter that is applied to \( \{e^{i(k-1)\omega_0} \tilde{y}_k(t)\}_{k=1}^L \). 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:

\[ h^* \tilde{a}_1 = 1 \]  

(6.5.110)

Using the above 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:

\[
\min_{h;\{s(t)\}} \sum_{t=1}^{N} \sum_{k=1}^{L} \left| h^* \tilde{y}_k(t) e^{i(k-1)\omega_0} - s(t) \right|^2 \quad \text{subject to:} \quad h^* \tilde{a}_1 = 1
\]  

(6.5.111)

The quadratic criterion in (6.5.111) expresses our desire to make the outputs of the spatial filter, \( \{h^* \tilde{y}_k(t) e^{i(k-1)\omega_0}\}_{k=1}^L \), resemble a signal \( s(t) \) (that is independent of \( k \)) as much as possible, in a least squares sense. Said another way, the above LS criterion expresses our goal to make the filter \( h \) attenuate any signal in \( \{\tilde{y}_k(t) e^{i(k-1)\omega_0}\}_{k=1}^L \), whose DOA is different from \( \theta \), 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

\[ g(t) = \frac{1}{L} \sum_{k=1}^{L} \tilde{y}_k(t) e^{i(k-1)\omega_0} \]  

(6.5.112)

and observe that

\[
\frac{1}{L} \sum_{k=1}^{L} \left| h^* \tilde{y}_k(t) e^{i(k-1)\omega_0} - s(t) \right|^2
= \left| s(t) \right|^2 + \frac{1}{L} \sum_{k=1}^{L} \left| \tilde{y}_k(t) \tilde{y}_k^*(t) \right| h - h^* g(t) s^*(t) - g^*(t) h s(t)
= h^* \left[ \frac{1}{L} \sum_{k=1}^{L} \tilde{y}_k(t) \tilde{y}_k^*(t) \right] h - h^* g(t) g^*(t) h + \left| s(t) - h^* g(t) \right|^2
\]  

(6.5.113)

Hence, the sequence \( \{s(t)\} \) that minimizes (6.5.111), for fixed \( h \), is given by

\[ \hat{s}(t) = h^* g(t) \]  

(6.5.114)
Chapter 6 Spatial Methods

Inserting (6.5.114) into (6.5.111) (see also (6.5.113)) we obtain the reduced problem:

$$\min_h h^* \hat{Q} h \quad \text{subject to: } h^* \tilde{a}_1 = 1$$

(6.5.115)

where

$$\hat{Q} = \hat{R} - \hat{G}$$

$$\hat{R} = \frac{1}{N} \sum_{t=1}^{N} \sum_{k=1}^{L} \bar{g}_k(t) \bar{g}_k^*(t)$$

(6.5.116)

$$\hat{G} = \frac{1}{N} \sum_{t=1}^{N} g(t) g^*(t)$$

The solution to the quadratic problem with linear constraints in (6.5.115) can be obtained by using Result R35 in Appendix A:

$$\hat{h} = \frac{\hat{Q}^{-1} \tilde{a}_1}{\tilde{a}_1^* \hat{Q}^{-1} \tilde{a}_1}$$

(6.5.117)

Using (6.5.117) in (6.5.114) we can obtain an estimate of the signal sequence, which may be of interest in some applications, as well as an estimate of the signal power:

$$\hat{\sigma}_s^2 = \frac{1}{N} \sum_{t=1}^{N} |\hat{s}(t)|^2 = \hat{h}^* \hat{G} \hat{h}$$

(6.5.118)

The above equation, as a function of DOA $\theta$, provides an estimate of the spatial spectrum.

The matrix $\hat{Q}$ in (6.5.116) can be rewritten in the following form:

$$\hat{Q} = \frac{1}{N} \sum_{t=1}^{N} \sum_{k=1}^{L} \left[ e^{i(k-1)\omega_s \bar{g}_k(t) - g(t)} \right] \left[ e^{i(k-1)\omega_s \bar{g}_k(t) - g(t)} \right]^*$$

(6.5.119)

It follows from (6.5.119) that $\hat{Q}$ is always positive semidefinite. For $L = 1$ (or, equivalently, $\tilde{m} = m$) we have $\hat{Q} = 0$ because $g(t) = \bar{g}_1(t)$ 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, $\tilde{m} \leq m - 1$). For $L \geq 2$ the $(\tilde{m} \times \tilde{m})$ matrix $\hat{Q}$ is a sum of $NL$ outer products; if $NL \geq \tilde{m}$, 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 $\tilde{m}$ is its maximum possible value:

$$\tilde{m} = m - 1 \quad \iff \quad L = 2$$

(6.5.120)
A numerical study of performance, reported in [Gini and Lombardini 2002], supports the above choice of \( \tilde{m} \), and also suggests that the spatial APES method may outperform the Capon method in both spatial spectrum estimation and DOA estimation applications. The APES spatial filter is, however, more difficult to compute than is the Capon spatial filter, owing to 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 \( \tilde{m} = m - 1 \) (and thus \( L = 2 \)) as in (6.5.120). Intuitively, this choice of \( \tilde{m} \) provides the APES filter with the maximum possible number of degrees of freedom, and hence it makes sense that it should lead to better resolution and interference rejection capability than would smaller values of \( \tilde{m} \). For \( L = 2 \) we have

\[
g(t) = \frac{1}{2} [\tilde{y}_1(t) + e^{j\omega_s} \tilde{y}_2(t)]
\]

and hence

\[
\hat{Q} = \frac{1}{2N} \sum_{t=1}^{N} \left\{ \frac{1}{4} [\tilde{y}_1(t) - e^{j\omega_s} \tilde{y}_2(t)]^* [\tilde{y}_1(t) - e^{j\omega_s} \tilde{y}_2(t)] + \frac{1}{4} [e^{j\omega_s} \tilde{y}_2(t) - \tilde{y}_1(t)]^* [e^{j\omega_s} \tilde{y}_2(t) - \tilde{y}_1(t)]^* \right\}
\]

\[
= \frac{1}{4N} \sum_{t=1}^{N} [\tilde{y}_1(t) - e^{j\omega_s} \tilde{y}_2(t)]^* [\tilde{y}_1(t) - e^{j\omega_s} \tilde{y}_2(t)]^*
\]

It follows that the APES spatial filter is the solution to the problem (see (6.5.115))

\[
\min_{h} \sum_{t=1}^{N} |h^* [\tilde{y}_1(t) - e^{j\omega_s} \tilde{y}_2(t)]|^2 \quad \text{subject to: } h^* \tilde{a}_1 = 1
\]

and that the APES signal estimate is given by (see (6.5.114))

\[
\hat{s}(t) = \frac{1}{2} h^* [\tilde{y}_1(t) + e^{j\omega_s} \tilde{y}_2(t)]
\]

On the other hand, the Capon spatial filter is obtained as the solution to the problem

\[
\min_{h} \sum_{t=1}^{N} |h^* y(t)|^2 \quad \text{subject to: } h^* a = 1
\]

and the Capon signal estimate is given by

\[
\hat{s}(t) = h^* y(t)
\]

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 \( \theta \), there is an interference signal \( i(t) \) that impinges on the array from another DOA, denoted \( \theta_i \). 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 similarly treated. The array output vector in (6.5.102) and the subvectors in (6.5.109) become

\[ y(t) = a(\theta)s(t) + b(\theta_i)i(t) + e(t) \]  
\[ y_1(t) = \tilde{a}_1(\theta)s(t) + \tilde{b}_1(\theta_i)i(t) + \tilde{e}_1(t) \]  
\[ y_2(t) = \tilde{a}_2(\theta)s(t) + \tilde{b}_2(\theta_i)i(t) + \tilde{e}_2(t) \]

where the quantities \( b, \tilde{b}_1, \) and \( \tilde{b}_2 \) are defined similarly to \( a, \tilde{a}_1, \) and \( \tilde{a}_2. \) We have shown the dependence of the various quantities on \( \theta \) and \( \theta_i \) in equations (6.5.127)–(6.5.129), but will drop the DOA dependence in the remainder of the derivation to simplify notation.

For the above scenario, the Capon method is known to have poor performance in either of the following two situations:

(i) The SOI steering vector is imprecisely known, for example owing to 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 \( |h^*a_0| \approx 0 \) (where \( \approx 0 \) 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 \( \rho. \) Then (6.5.127) can be rewritten as

\[ y(t) = (a + \rho b)s(t) + e(t) \]  

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 \( |h^*(a + \rho b)| \approx 0, \) 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

\[
\begin{align*}
[\tilde{y}_1(t) - e^{j\omega_s}\tilde{y}_2(t)] &= (\tilde{a}_1 - e^{j\omega_s}\tilde{a}_2) s(t) + (\tilde{b}_1 - e^{j\omega_s}\tilde{b}_2) i(t) + [\tilde{e}_1(t) - e^{j\omega_s}\tilde{e}_2(t)] \\
&= \left[ 1 - e^{j(\omega_s - \omega_i)} \right] \tilde{b}_1 i(t) + [\tilde{e}_1(t) - e^{j\omega_s}\tilde{e}_2(t)]
\end{align*}
\]
and

\[
\frac{1}{2} \left( \bar{y}_1(t) + e^{i\omega_s} \bar{y}_2(t) \right) \\
= \frac{1}{2} \left( \bar{a}_1 + e^{i\omega_s} \bar{a}_2 \right) s(t) + \frac{1}{2} \left( \bar{b}_1 + e^{i\omega_s} \bar{b}_2 \right) i(t) + \frac{1}{2} \left[ \bar{e}_1(t) + e^{i\omega_s} \bar{e}_2(t) \right]
\]

\[
= \bar{a}_1 s(t) + \frac{1}{2} \left[ 1 + e^{i(\omega_s - \omega_j)} \right] \bar{b}_1 i(t) + \frac{1}{2} \left[ \bar{e}_1(t) + e^{i\omega_s} \bar{e}_2(t) \right]
\]  

(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

\[ |1 - e^{i(\omega_s - \omega_j)}| \cdot |h^* \bar{b}_1| \simeq 0 \]  

(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 similarly to the Capon method. 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), and 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:** A similar argument 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

\[
\bar{y}_1(t) - e^{i\omega_s} \bar{y}_2(t) = \left[ 1 - e^{i(\omega_s - \omega_s^0)} \right] \bar{a}_1^0 s(t) + \left[ 1 - e^{i(\omega_s - \omega_s^0)} \right] \bar{b}_1 i(t) + \left[ \bar{e}_1(t) - e^{i\omega_s} \bar{e}_2(t) \right]
\]  

(6.5.134)

It follows that in case (i) the APES spatial filter tends to cancel the SOI as well, in addition to cancelling the interference. However, the pointing errors are usually quite small, and therefore the residual term of \( s(t) \) in (6.5.134) is small as well. Hence, the SOI may 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 above 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 explained above.

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 to 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, which may allow the second factor to increase somewhat. This explains why the beamwidth of the APES spatial filter may be larger than that of the Capon filter, and hence why APES may have a slightly lower resolution.

### 6.5.7 The CLEAN Algorithm

The CLEAN algorithm is a semi-parametric method that can be used for spatial spectral estimation. As we will see, this algorithm can be introduced in a non-parametric fashion (see [Högbohm 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 semi-parametric approach.

There is a significant literature about CLEAN and its many applications in diverse areas, including array signal processing, image processing, and astronomy (see, e.g., [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):

$$\hat{\phi}_1(\theta) = a^*(\theta) \hat{R} a(\theta)$$  \hspace{1cm} (6.5.135)

where $a(\theta)$ and $\hat{R}$ are defined as in Section 6.3.1. Let

$$\hat{\theta}_1 = \arg \max_{\theta} \hat{\phi}_1(\theta)$$  \hspace{1cm} (6.5.136)

$$\hat{\sigma}_1^2 = \frac{1}{m^2} \hat{\phi}_1(\hat{\theta}_1)$$  \hspace{1cm} (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 by (see, e.g., (6.2.19)):

$$\hat{\sigma}_1^2 a(\hat{\theta}_1) a^*(\hat{\theta}_1) \quad (6.5.138)$$

Consequently, the expected term in $\hat{\phi}_1(\theta)$ due to (6.5.138) is

$$\sigma_1^2 \left| a^*(\theta) a(\hat{\theta}_1) \right|^2 \quad (6.5.139)$$

We partly eliminate the term (6.5.139) from $\hat{\phi}_1(\theta)$, and hence define a new spectrum

$$\hat{\phi}_2(\theta) = \hat{\phi}_1(\theta) - \rho\hat{\sigma}_1^2 \left| a^*(\theta) a(\hat{\theta}_1) \right|^2 \quad (6.5.140)$$

where $\rho$ is a user parameter that satisfies

$$\rho \in (0, 1] \quad (6.5.141)$$

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), may not necessarily be true. For example, there may 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 may 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 may 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, similarly to what we said in (a) above.

(c) In both situations above, 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 $\rho = 1$) may yield a spatial spectrum that takes on negative values at some DOAs (which it should not). Using $\rho < 1$ in (6.5.140) reduces the likelihood that this undesirable event happens too early in the iterative process of the CLEAN algorithm (see below).
Chapter 6 Spatial Methods

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

$$\hat{\theta}_2 = \arg \max_\theta \hat{\phi}_2(\theta)$$

$$\sigma_2^2 = \frac{1}{m^2} \hat{\phi}_2(\hat{\theta}_2)$$

and

$$\hat{\phi}_3(\theta) = \hat{\phi}_2(\theta) - \rho \sigma_2^2 |a^*(\theta) a(\hat{\theta}_2)|^2$$

Continuing the iterations in the same manner as above yields the CLEAN algorithm, a compact description of which is as follows:

The CLEAN Algorithm

Initialization: $\hat{\phi}_1(\theta) = a^*(\theta) \hat{R}(\theta)$

For $k = 1, 2, \ldots$ do:

$$\hat{\theta}_k = \arg \max_\theta \hat{\phi}_k(\theta)$$

$$\sigma_k^2 = \frac{1}{m^2} \hat{\phi}_k(\hat{\theta}_k)$$

$$\hat{\phi}_{k+1}(\theta) = \hat{\phi}_k(\theta) - \rho \sigma_k^2 |a^*(\theta) a(\hat{\theta}_k)|^2$$

We continue the iterative process in (6.5.145) until either we complete a prespecified number of iterations or until $\hat{\phi}_k(\theta)$ for some $k$ has become (too) negative at some DOAs (see, e.g., [Hogbom 1974; Cornwell and Bridle 1996]).

Regarding the choice of $\rho$ in (6.5.145), while 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., [Hogbom 1974; Cornwell and Bridle 1996; Schwarz 1978b]). We will make further comments on the choice of $\rho$ 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 due to its mainlobe smearing and sidelobe leakage problems. The discrete spatial spectral estimate $\{\rho \sigma_k^2, \hat{\theta}_k\}_{k=1,2,\ldots}$ 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 above 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 semi-parametric for short.

Next, we present a more formal derivation of the CLEAN algorithm. Consider the following semi-parametric model of the array output covariance matrix

$$R = \sigma_1^2 a(\theta_1) a^*(\theta_1) + \sigma_2^2 a(\theta_2) a^*(\theta_2) + \cdots \quad (6.5.146)$$

As implied by the previous discussion, this is the covariance model assumed by CLEAN. Let us fit (6.5.146) to the sample covariance matrix $\hat{R}$ in a least squares sense:

$$\min_{\{\sigma_1^2, \sigma_2^2, \ldots\}} \left\| \hat{R} - \sigma_1^2 a(\theta_1) a^*(\theta_1) - \sigma_2^2 a(\theta_2) a^*(\theta_2) - \cdots \right\|^2 \quad (6.5.147)$$

We will show that CLEAN is a sequential algorithm for approximately minimizing the above 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 $(\sigma_1^2, \theta_1)$ by minimizing (6.5.147) with $\sigma_2^2 = + \cdots = 0$:

$$\min_{\sigma_1^2, \theta_1} \left\| \hat{R} - \sigma_1^2 a(\theta_1) a^*(\theta_1) \right\|^2 \quad (6.5.148)$$

As shown in Complement 6.5.3, the solution to (6.5.148) is given by

$$\hat{\theta}_1 = \arg \max_\theta \hat{\phi}_1(\theta); \quad \hat{\sigma}_1^2 = \frac{1}{m^2} \hat{\phi}_1(\hat{\theta}_1) \quad (6.5.149)$$

where $\hat{\phi}_1(\theta)$ is as defined previously. We reduce the above power estimate by using $\sigma_1^2$ in lieu of $\sigma_1^2$. The reasons for this reduction are discussed in points (a)–(c) above; in particular, we would like the residual covariance matrix $\hat{R} - \rho \sigma_1^2 a(\hat{\theta}_1) a^*(\hat{\theta}_1)$ 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 $(\sigma_1^2, \theta_2)$ by minimizing (6.5.147) with $\sigma_1^2 = \rho \sigma_1^2, \theta_1 = \hat{\theta}_1$ and $\sigma_2^2 = \sigma_3^2 = \cdots = 0$:

$$\min_{\sigma_1^2, \theta_2} \left\| \hat{R} - \rho \sigma_1^2 a(\hat{\theta}_1) a^*(\hat{\theta}_1) - \sigma_2^2 a(\theta_2) a^*(\theta_2) \right\|^2 \quad (6.5.150)$$

The solution to (6.5.150) can be shown to be (similarly to solving (6.5.148)):

$$\hat{\theta}_2 = \arg \max_\theta \hat{\phi}_2(\theta); \quad \hat{\sigma}_2^2 = \frac{1}{m^2} \hat{\phi}_2(\hat{\theta}_2) \quad (6.5.151)$$

where

$$\hat{\phi}_2(\theta) = a^*(\theta) \left[ \hat{R} - \rho \sigma_1^2 a(\hat{\theta}_1) a^*(\hat{\theta}_1) \right] a(\theta)$$

$$= \hat{\phi}_1(\theta) - \rho \sigma_1^2 \left| a^*(\theta) a(\hat{\theta}_1) \right|^2 \quad (6.5.152)$$
Chapter 6  Spatial Methods

Observe that (6.5.149) and (6.5.151) coincide with (6.5.136)–(6.5.137) and (6.5.142)–(6.5.143). Evidently, continuing the above iterative process, for which (6.5.149) and (6.5.151) are the first two steps, leads to the CLEAN algorithm in (6.5.145).

The above derivation of CLEAN sheds some light on the properties of this algorithm. First, note that the LS covariance fitting criterion in (6.5.147) is decreased at each iteration of CLEAN. For instance, consider the first iteration. A straightforward calculation shows that:

\[ \| \hat{R} - \rho \hat{\sigma}_1^2 a(\hat{\theta}_1) a^*(\hat{\theta}_1) \|^2 \\
= \| \hat{R} \|^2 - 2 \rho \hat{\sigma}_1^2 a^*(\hat{\theta}_1) \hat{R} a(\hat{\theta}_1) + \rho^2 \hat{\sigma}_1^4 \\
= \| \hat{R} \|^2 - \rho (2 - \rho) \hat{\sigma}_1^4 \]  

(6.5.153)

Clearly, (6.5.153) 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 (6.5.147) 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 (6.5.147) as much and as fast as possible, but also to ensure that the residual covariance matrices

\[ \hat{R}_{k+1} = \hat{R}_k - \rho \hat{\sigma}_k^2 a(\hat{\theta}_k) a^*(\hat{\theta}_k) ; \quad \hat{R}_1 = \hat{R} \]  

(6.5.154)

remain positive definite for \( k = 1, 2, \ldots \); otherwise, fitting \( \sigma_k^2 a(\theta_{k+1}) a^*(\theta_{k+1}) \) 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

\[ \rho < \frac{1}{\hat{\sigma}_k^2 a^*(\hat{\theta}_k) \hat{R}_k^{-1} a(\hat{\theta}_k)} \]  

(6.5.155)

Note that the right-hand side of (6.5.155) is bounded above by one, because by the Cauchy–Schwartz inequality:

\[ \hat{\sigma}_k^2 a^*(\hat{\theta}_k) \hat{R}_k^{-1} a(\hat{\theta}_k) = \frac{1}{m^2} \left[ a^*(\hat{\theta}_k) \hat{R}_k a(\hat{\theta}_k) \right] \left[ a^*(\hat{\theta}_k) \hat{R}_k^{-1} a(\hat{\theta}_k) \right] \\
= \frac{1}{m^2} \| \hat{R}_k^{1/2} a(\hat{\theta}_k) \|^2 \| \hat{R}_k^{-1/2} a(\hat{\theta}_k) \|^2 \\
\geq \frac{1}{m^2} \| a^*(\hat{\theta}_k) \hat{R}_k^{1/2} \hat{R}_k^{-1/2} a(\hat{\theta}_k) \|^2 \\
= \frac{1}{m^2} \| a^*(\hat{\theta}_k) a(\hat{\theta}_k) \|^2 = 1 \]

Also note that, depending on the scenario under consideration, satisfaction of the inequality in (6.5.155) for \( k = 1, 2, \ldots \) may require choosing a value for \( \rho \) much less than one. In summary, the above discussion has provided 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 semi-parametric nature of CLEAN more transparent. Specifically, the discussion has shown that
CLEAN fits the semi-parametric covariance model in (6.5.146) 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 (6.5.147) may also be useful to understand 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 $\{y(t)\}_{t=1,2,\ldots}$ 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

$$p_y(y(t)) = \frac{1}{\pi^m |R|} e^{-y^T(t)R^{-1}y(t)}$$  \hspace{1cm} (6.5.156)

Assume that $N$ observations of $\{y(t)\}$ are available:

$$\{y(1), \ldots, y(N)\}$$  \hspace{1cm} (6.5.157)

Owing to the i.i.d. assumption made on the sequence $\{y(t)\}_{t=1,2,\ldots}$ the probability density function of the sample in (6.5.157) is given by:

$$p(y(1), \ldots, y(N)) = \prod_{t=1}^{N} p(y(t)) = \frac{1}{\pi^m N |R|^N} e^{-\sum_{i=1}^{N} y^T(t)R^{-1}y(t)}$$  \hspace{1cm} (6.5.158)

The maximum likelihood (ML) estimate of the covariance matrix $R$, based on the sample in (6.5.157), is given by the maximizer of the likelihood function in (6.5.158) (see Section B.1 in Appendix B) or, equivalently, by the minimizer of the negative log-likelihood function:

$$- \ln p(y(1), \ldots, y(N)) = mN \ln(\pi) + N \ln |R| + \sum_{t=1}^{N} y^T(t)R^{-1}y(t)$$  \hspace{1cm} (6.5.159)

The part of (6.5.159) that depends on $R$ is given by (after multiplying by $\frac{1}{N}$)

$$\ln |R| + \frac{1}{N} \sum_{t=1}^{N} y^T(t)R^{-1}y(t) = \ln |R| + \text{tr} \left( R^{-1} \hat{R} \right)$$  \hspace{1cm} (6.5.160)

where

$$\hat{R} = \frac{1}{N} \sum_{t=1}^{N} y(t)y^T(t) \quad (m \times m)$$  \hspace{1cm} (6.5.161)
In this complement we discuss the minimization of (6.5.160) 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

$$JR^TJ = R \iff R = \frac{1}{2}(R + JR^TJ) \quad (6.5.162)$$

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 (6.5.161),

$$\hat{R}_{U,ML} = \hat{R} \quad (6.5.163)$$

whereas the persymmetric ML estimate of $R$, denoted $\hat{R}_{P,ML}$, is given by

$$\hat{R}_{P,ML} = \frac{1}{2} (\hat{R} + J\hat{R}^TJ) \quad (6.5.164)$$

To prove (6.5.163) we need to show that (see (6.5.160)):

$$\ln |R| + \text{tr} \left( R^{-1}\hat{R} \right) \geq \ln |\hat{R}| + m \quad \text{for any } R > 0 \quad (6.5.165)$$

Let $\hat{C}$ be a square root of $\hat{R}$ (see Definition D12 in Appendix A) and note that

$$\text{tr} \left( R^{-1}\hat{R} \right) = \text{tr} \left( R^{-1}\hat{C}\hat{C}^* \right) = \text{tr} \left( \hat{C}^*R^{-1}\hat{C} \right) \quad (6.5.166)$$

Using (6.5.166) in (6.5.165) we obtain the following series of equivalences:

(6.5.165) $\iff$ tr $\left( \hat{C}^*R^{-1}\hat{C} \right) - \ln |R^{-1}\hat{R}| \geq m$

$\iff$ tr $\left( \hat{C}^*R^{-1}\hat{C} \right) - \ln |\hat{C}^*R^{-1}\hat{C}| \geq m$

$\iff$ $\sum_{k=1}^{m} (\lambda_k - \ln \lambda_k - 1) \geq 0 \quad (6.5.167)$

where $\{\lambda_k\}$ are the eigenvalues of the matrix $\hat{C}^*R^{-1}\hat{C}$.
Next we show, with reference to (6.5.167), that

\[
\begin{align*}
\frac{d}{d\lambda} \ln \lambda - \frac{1}{\lambda} = \lambda - \ln \lambda - 1 & \geq 0 \quad \text{for any } \lambda > 0 \\
\frac{d^2}{d\lambda^2} \ln \lambda - \frac{1}{\lambda^2} = \frac{1}{\lambda^2} & > 0
\end{align*}
\]

To verify (6.5.168), observe that

\[
\begin{align*}
f'(\lambda) &= 1 - \frac{1}{\lambda}, \\
f''(\lambda) &= \frac{1}{\lambda^2}
\end{align*}
\]

Hence, the function \( f(\lambda) \) in (6.5.168) has a unique minimum at \( \lambda = 1 \), and \( f(1) = 0 \); this proves (6.5.168). With this observation, the proof of (6.5.167), and therefore of (6.5.163), is complete.

The proof of (6.5.164) is even simpler. In view of (6.5.162), we have that

\[
\begin{align*}
\text{tr} \left( \hat{R}^{-1} \hat{R} \right) &= \text{tr} \left[ \left( J \hat{R}^T J \right)^{-1} \hat{R} \right] \\
&= \text{tr} \left( \hat{R}^{-1} J \hat{R}^T J \right)
\end{align*}
\]

Hence, the function to be minimized with respect to \( R \) (under the constraint (6.5.162)) can be written as:

\[
\ln |R| + \text{tr} \left[ R^{-1} : \frac{1}{2} \left( \hat{R} + J \hat{R}^T J \right) \right]
\]

As shown earlier in this complement, the unstructured minimizer of (6.5.170) is given by

\[
R = \frac{1}{2} \left( \hat{R} + J \hat{R}^T J \right)
\]

Because (6.5.171) satisfies the persymmetry constraint, by construction, it also gives the constrained minimizer of the negative log-likelihood function, and hence the proof of (6.5.164) 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 using a Sensor in Motion

This exercise illustrates how the directions of arrival of planar waves can be determined 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 \( \theta_k \) denote the \( k \)th 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 \( \{ \alpha_k e^{j \omega t} \} \) with the same (center) frequency \( \omega \). These signals may be reflections of a probing