6

Spatial Methods

6.1 INTRODUCTION

In this chapter, we consider the problem of *locating n radiating sources by using an array of m passive sensors*, as shown in Figure 6.1. The emitted energy from the sources could be, for example, acoustic or electromagnetic, and the receiving sensors could be any transducers that convert the received energy to electrical signals. Examples of sensors include electromagnetic antennas, hydrophones, and seismometers. This type of problem finds applications in *radar and sonar systems, communications, astrophysics, biomedical research, seismology, underwater surveillance* (also called passive listening), and many other fields. This problem basically consists of determining how the "energy" is distributed over *space* (which can be air, water, or the earth), with the source positions representing points in space with high concentrations of energy. Hence, it can be named a *spatial spectral estimation problem*. This name is also motivated by the fact that there are close ties between the source-location problem and the problem of temporal spectral estimation, treated in Chapters 1–5. In fact, as we will see, almost any of the methods encountered in the previous chapters may be used to derive a solution for the source-location problem.

The emphasis in this chapter will be on *developing a model for the output signal of the receiving sensor array*. Once this model has been derived, the source-location problem is turned into a parameter-estimation problem that is quite similar to the temporal frequency-finding application discussed in Chapter 4. Hence, as we shall see, most of the methods developed for frequency estimation can be used to solve the spatial problem of source location.

The sources in Figure 6.1 generate a *wave field* that travels through space and is *sampled, in both space and time, by the sensor array*. By making an analogy with temporal sampling, we may expect that the spatial sampling done by the array provides more and more information on

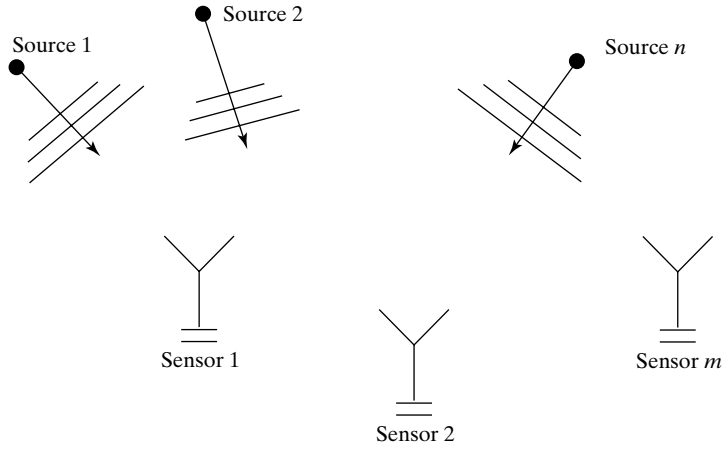


Figure 6.1 The set-up of the source-location problem.

the incoming waves as the *array's aperture* increases. The array's aperture is the space occupied by the array, as measured in units of signal wavelength. It is then no surprise that an array of sensors can provide significantly enhanced location performance as compared to the use of a *single antenna* (which was the system used in the early applications of the source-location problem.)

The development of the array model in the next section is based on a number of simplifying assumptions. Some of these assumptions, which have a more general character, are listed below. The sources are assumed to be situated in the *far field* of the array. Furthermore, we assume that both the sources and the sensors in the array are in the *same plane* and that the sources are *point emitters*. In addition, it is assumed that the *propagation medium is homogeneous (*i.e.*, not dispersive)*, and so the waves arriving at the array can be considered to be *planar*. Under these assumptions, the only parameter that characterizes the source locations is the so-called *angle of arrival*, or *direction of arrival* (DOA); the DOA will be formally defined later on.

The above assumptions may be relaxed, but only at the expense of significantly complicating the array model. Note that, in the general case of a near-field source and a three-dimensional array, three parameters are required to define the position of one source—for instance the *azimuth*, *elevation*, and *range*. Nevertheless, if the assumption of planar waves is maintained, then we can treat the case of several unknown parameters per source without complicating the model too much. However, in order to keep the discussion as simple as possible, we will consider only the case of one parameter per source.

In this chapter, it is also assumed that *the number of sources n is known*. The selection of n, when it is unknown, is a problem of significant importance for many applications, which is often referred to as the *detection problem*. For solutions to the detection problem (which is analogous to the problem of order selection in signal modeling), the reader is referred to [Wax and Kailath 1985; Fuchs 1988; Viberg, Ottersten, and Kailath 1991; Fuchs 1992] and Appendix C.

Finally, it is assumed that the sensors in the array can be modeled as linear (time-invariant) systems, and that both their transfer characteristics and their locations are known. In short, we say that *the array is assumed to be calibrated*.

6.2 ARRAY MODEL

We begin by considering the case of a *single source*. Once we establish a model of the array for this case, the general model for the multiple-source case is simply obtained by the superposition principle.

Suppose that a single waveform impinges upon the array, and let $x(t)$ denote the value of the signal waveform as measured at some *reference point*, at time t. The "reference point" may be one of the sensors in the array or any other point placed near enough to the array so that the previously made assumption of planar wave propagation holds true. The physical signals received by the array are *continuous time waveforms*; hence, t is a continuous variable here, unless otherwise stated.

Let τ_k denote the time needed for the wave to travel from the reference point to sensor k ($k = 1, \ldots, m$). Then the output of sensor k can be written as

$$\bar{y}_k(t) = \bar{h}_k(t) * x(t - \tau_k) + \bar{e}_k(t) \tag{6.2.1}$$

where $\bar{h}_k(t)$ is the impulse response of the kth sensor, "$*$" denotes the convolution operation, and $\bar{e}_k(t)$ is an *additive noise*. The noise may enter in equation (6.2.1) either as "thermal noise" generated by the sensor's circuitry, as "random background radiation" impinging on the array, or in other ways. In (6.2.1), $\bar{h}_k(t)$ is assumed known, but both the "input" signal $x(t)$ and the delay τ_k are unknown. The parameters characterizing the source location enter into (6.2.1) through $\{\tau_k\}$. Hence, the source-location problem is basically one of *time-delay estimation for the unknown input case*.

The model equation (6.2.1) can be simplified significantly if *the signals are assumed to be narrowband*. In order to show how this can be done, a number of preliminaries are required.

Let $X(\omega)$ denote the Fourier transform of the (continuous-time) signal $x(t)$:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt \tag{6.2.2}$$

(which is assumed to exist and be finite for all $\omega \in (-\infty, \infty)$). The inverse transform, which expresses $x(t)$ as a linear functional of $X(\omega)$, is given by

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{i\omega t} d\omega \tag{6.2.3}$$

Similarly, we define the transfer function $\bar{H}_k(\omega)$ of the kth sensor as the Fourier transform of $\bar{h}_k(t)$. In addition, let $\bar{Y}_k(\omega)$ and $\bar{E}_k(\omega)$ denote the Fourier transforms of the signal $\bar{y}_k(t)$ and noise $\bar{e}_k(t)$ in (6.2.1). By using this notation and the properties of the Fourier transform, $\bar{Y}_k(\omega)$ can be written as

$$\bar{Y}_k(\omega) = \bar{H}_k(\omega) X(\omega) e^{-i\omega\tau_k} + \bar{E}_k(\omega) \tag{6.2.4}$$

For a general class of physical signals, such as the carrier-modulated signals encountered in communications, the energy spectral density of $x(t)$ has the form shown in Figure 6.2. There, ω_c denotes the *center (or carrier) frequency*, which is usually the center of the frequency band occupied by the signal (hence its name). A signal having an energy spectrum of the form depicted in Figure 6.2 is called a *bandpass signal* (by direct analogy with the notion of bandpass filters).

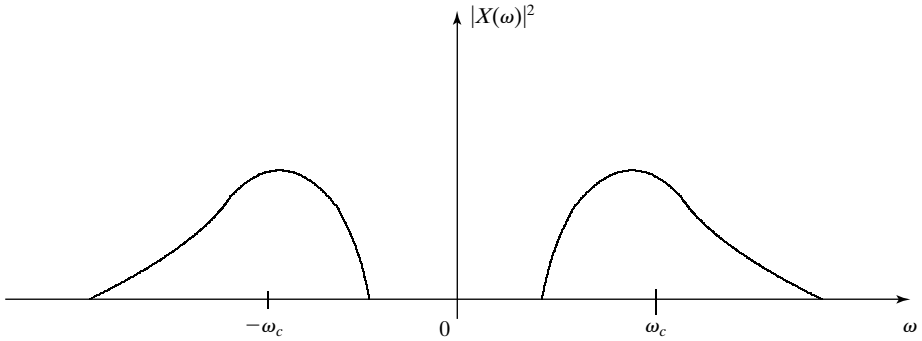


Figure 6.2 The energy spectrum of a bandpass signal.

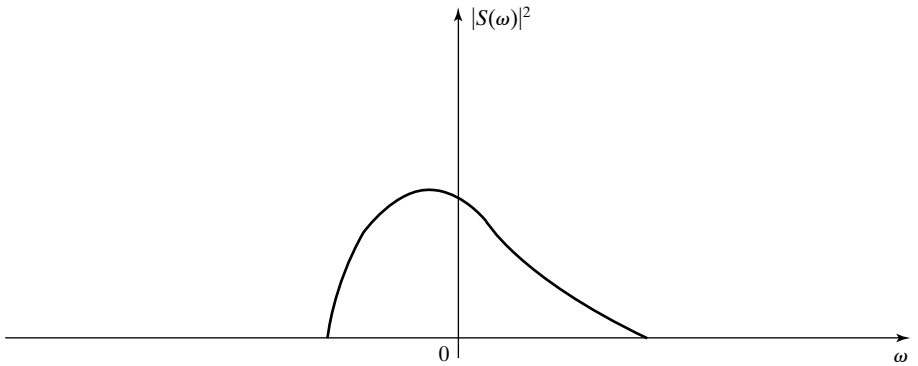


Figure 6.3 The baseband spectrum that gives rise to the bandpass spectrum in Figure 6.2.

For now, *we assume that the received signal $x(t)$ is bandpass*. It is clear from Figure 6.2 that the spectrum of such a signal is completely defined by the spectrum of a corresponding *baseband (or lowpass) signal*. The baseband spectrum—say, $|S(\omega)|^2$—corresponding to the one in Figure 6.2, is displayed in Figure 6.3. Let $s(t)$ denote the baseband signal associated with $x(t)$. The process of obtaining $x(t)$ from $s(t)$ is called *modulation*; the inverse process is called *demodulation*. In what follows, we make a number of comments on the modulation and demodulation processes, which—while not being strictly relevant to the source-location problem—could be helpful in clarifying some claims in the text.

6.2.1 The Modulation-Transmission-Demodulation Process

The physical signal $x(t)$ is real valued; hence, its spectrum $|X(\omega)|^2$ should be even (i.e., symmetric about $\omega = 0$; see, for instance, Figure 6.2). On the other hand, the spectrum of the demodulated signal $s(t)$ might not be even (as indicated in Figure 6.3); hence, $s(t)$ might be complex valued. The way in which this can happen is explained as follows. The *transmitted* signal is, of course,

obtained by modulating a real-valued signal. Hence, in the spectrum of the transmitted signal, the baseband spectrum is symmetric about $\omega = \omega_c$. The characteristics of the *transmission channel* (or the *propagation medium*), however, most often are asymmetric about $\omega = \omega_c$. This results in a *received* bandpass signal with an associated baseband spectrum that is not even. Hence, the demodulated received signal is complex-valued. This observation supports a claim made in Chapter 1 that complex-valued signals are not uncommon in spectral estimation problems.

The Modulation Process. If $s(t)$ is multiplied by $e^{i\omega_c t}$, then the Fourier transform of $s(t)$ is translated in frequency to the right by ω_c (assumed to be positive), as is verified by

$$\int_{-\infty}^{\infty} s(t)e^{i\omega_c t}e^{-i\omega t}d\omega = \int_{-\infty}^{\infty} s(t)e^{-i(\omega-\omega_c)t}d\omega = S(\omega-\omega_c) \tag{6.2.5}$$

This formula describes the essence of the so-called *complex modulation process*. (An analogous formula for random discrete-time signals is given by equation (1.4.11) in Chapter 1.) The output of the complex modulation process is always complex valued (hence the name of this form of modulation). If the modulated signal is real valued, as $x(t)$ is, then it must have an even spectrum. In such a case, the translation of $S(\omega)$ to the right by ω_c, as in (6.2.5), must be accompanied by a translation to the left (also by ω_c) of the folded and complex-conjugated baseband spectrum. This process results in the following expression for $X(\omega)$:

$$\boxed{X(\omega) = S(\omega-\omega_c) + S^*(-(\omega+\omega_c)\,)} \tag{6.2.6}$$

It is readily verified that, in the time domain, the *real modulation process* leading to (6.2.6) corresponds to taking the real part of the complex-modulated signal $s(t)e^{i\omega_c t}$; that is,

$$\begin{aligned} x(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}[S(\omega-\omega_c)+S^*(-\omega-\omega_c)]e^{i\omega t}d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega-\omega_c)e^{i(\omega-\omega_c)t}e^{i\omega_c t}d\omega \\ &\quad + \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} S(-\omega-\omega_c)e^{-i(\omega+\omega_c)t}e^{i\omega_c t}d\omega\right]^* \\ &= s(t)e^{i\omega_c t} + [s(t)e^{i\omega_c t}]^* \end{aligned}$$

which gives

$$\boxed{x(t) = 2\mathrm{Re}[s(t)e^{i\omega_c t}]} \tag{6.2.7}$$

or

$$\boxed{x(t) = 2\alpha(t)\cos(\omega_c t + \varphi(t))} \tag{6.2.8}$$

where $\alpha(t)$ and $\varphi(t)$ are the amplitude and phase of $s(t)$, respectively:

$$s(t) = \alpha(t)e^{i\varphi(t)}$$

If we let $s_I(t)$ and $s_Q(t)$ denote the real and imaginary parts of $s(t)$, then we can also write (6.2.7) as

$$\boxed{x(t) = 2[s_I(t)\cos(\omega_c t) - s_Q(t)\sin(\omega_c t)]} \tag{6.2.9}$$

We note in passing the following terminology associated with the equivalent time-domain representations (6.2.7)–(6.2.9) of a bandpass signal: $s(t)$ is called *the complex envelope* of $x(t)$, and $s_I(t)$ and $s_Q(t)$ are said to be *the in-phase and quadrature components* of $x(t)$.

The Demodulation Process. A calculation similar to (6.2.5) shows that the Fourier transform of $x(t)e^{-i\omega_c t}$ is given by

$$[S(\omega) + S^*(-\omega - 2\omega_c)]$$

which is simply $X(\omega)$ translated in frequency to the left by ω_c. The baseband (or lowpass) signal $s(t)$ can then be obtained by filtering $x(t)e^{-i\omega_c t}$ with a *baseband (or lowpass) filter* whose bandwidth is matched to that of $S(\omega)$. The hardware implementation of the demodulation process in block diagram form is presented in Figure 6.4.

6.2.2 Derivation of the Model Equation

Given the background of the previous subsection, we return to equation (6.2.4) describing the output of sensor k. Since $x(t)$ is assumed to be a bandpass signal, $X(\omega)$ is given by (6.2.6) which, when inserted in (6.2.4), leads to

$$\bar{Y}_k(\omega) = \bar{H}_k(\omega)[S(\omega - \omega_c) + S^*(-\omega - \omega_c)]e^{-i\omega\tau_k} + \bar{E}_k(\omega) \tag{6.2.10}$$

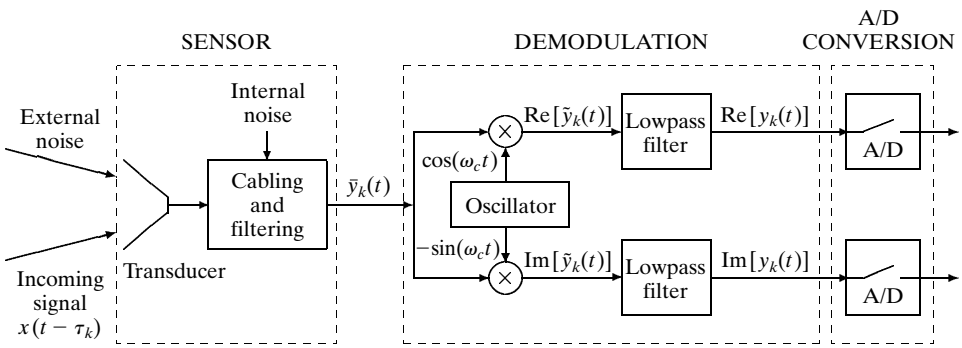


Figure 6.4 A simplified block diagram of the analog processing in a receiving-array element.

Let $\tilde{y}_k(t)$ denote the demodulated signal:

$$\tilde{y}_k(t) = \bar{y}_k(t)e^{-i\omega_c t}$$

It follows from (6.2.10) and the previous discussion on the demodulation process that the Fourier transform of $\tilde{y}_k(t)$ is given by

$$\begin{aligned}\tilde{Y}_k(\omega) = \bar{H}_k(\omega+\omega_c)[S(\omega) + S^*(-\omega-2\omega_c)]e^{-i(\omega+\omega_c)\tau_k} \\ + \bar{E}_k(\omega+\omega_c)\end{aligned} \tag{6.2.11}$$

When $\tilde{y}_k(t)$ is passed through a lowpass filter with bandwidth matched to $S(\omega)$, in the filter output (say, $y_k(t)$) the component in (6.2.11) centered at $\omega = -2\omega_c$ is eliminated along with all the other frequency components that fall in the stopband of the lowpass filter. Hence, we obtain

$$Y_k(\omega) = H_k(\omega+\omega_c)S(\omega)e^{-i(\omega+\omega_c)\tau_k} + E_k(\omega+\omega_c) \tag{6.2.12}$$

where $H_k(\omega+\omega_c)$ and $E_k(\omega+\omega_c)$ denote the parts of $\bar{H}_k(\omega+\omega_c)$ and $\bar{E}_k(\omega+\omega_c)$ that fall within the lowpass filter's passband, Ω, and where the frequency ω is restricted to Ω.

We now make the following *key assumption*:

The received signals are narrowband, so that $\lvert S(\omega)\rvert$ decreases rapidly with increasing $\lvert\omega\rvert$.	(6.2.13)

Under this assumption, (6.2.12) reduces (in an approximate way) to the following equation:

$$Y_k(\omega) = H_k(\omega_c)S(\omega)e^{-i\omega_c\tau_k} + E_k(\omega+\omega_c) \qquad \text{for } \omega \in \Omega \tag{6.2.14}$$

Because $H_k(\omega_c)$ must be different from zero, the sensor transfer function $\bar{H}_k(\omega)$ should pass frequencies near $\omega = \omega_c$ (as expected, since ω_c is the center frequency of the received signal). Also, note that we do not replace $E_k(\omega+\omega_c)$ in (6.2.14) by $E_k(\omega_c)$, as this term might not be (nearly) constant over the signal bandwidth. (For instance, this would be the case when the noise term in (6.2.12) contains a narrowband interference with the same center frequency as the signal.)

Remark: It is sometimes claimed that (6.2.12) can be reduced to (6.2.14) even if the *signals are broadband*, but *the sensors in the array are narrowband* with center frequency $\omega = \omega_c$. Under such an assumption, $|H_k(\omega+\omega_c)|$ goes quickly to zero as $|\omega|$ increases; hence, (6.2.12) becomes

$$Y_k(\omega) = H_k(\omega+\omega_c)S(0)e^{-i\omega_c\tau_k} + E_k(\omega+\omega_c) \tag{6.2.15}$$

which apparently is different from (6.2.14). In order to obtain (6.2.14) from (6.2.12) under the previous conditions, we need to make some additional assumptions. Hence, if we further assume that *the sensor frequency response is flat over the passband* (so that $H_k(\omega+\omega_c) = H_k(\omega_c)$) and

that *the signal spectrum varies over the sensor passband* (so that $S(\omega)$ differs quite a bit from $S(0)$ over the passband in question), then we can still obtain (6.2.14) from (6.2.12). ■

The model of the array is derived in a straightforward manner from equation (6.2.14). The time-domain counterpart of (6.2.14) is the following:

$$y_k(t) = H_k(\omega_c)e^{-i\omega_c\tau_k}s(t) + e_k(t) \tag{6.2.16}$$

where $y_k(t)$ and $e_k(t)$ are the inverse Fourier transforms of the corresponding terms in (6.2.14). (By a slight abuse of notation, $e_k(t)$ is associated with $E_k(\omega + \omega_c)$, not $E_k(\omega)$.)

The hardware implementation required to obtain $\{y_k(t)\}$, as defined above, is indicated in Figure 6.4. Note that the scheme in Figure 6.4 generates samples of the real and imaginary components of $y_k(t)$. These samples are paired in the digital machine following the analog scheme of Figure 6.4 to obtain samples of the complex-valued signal $y_k(t)$. (We stress once more that all physical analog signals are real valued.) Note that the continuous-time signal in (6.2.16) is *bandlimited*: According to (6.2.14) (and the related discussion), $Y_k(\omega)$ is approximately equal to zero for $\omega \notin \Omega$. Here Ω is the support of $S(\omega)$ (recall that the filter bandwidth is matched to the signal bandwidth); hence, it is a narrow interval. Consequently, we can sample (6.2.16) with a rather low sampling frequency.

The sampled version of $\{y_k(t)\}$ is used by the "digital processing equipment" for the purpose of DOA estimation. Of course, the *digital form* of $\{y_k(t)\}$ satisfies an equation directly analogous to (6.2.16). In fact, to avoid a complication of notation by the introduction of a new discrete-time variable, *from here on we consider that t in equation (6.2.16) takes discrete values*:

$$t = 1, 2, \ldots, N \tag{6.2.17}$$

(As usual, we choose the sampling period as the unit of the time axis.) In Figure 6.4 we sample the baseband signal, which may be done by using sampling rates lower than those needed for the bandpass signal. (See also [Proakis, Rader, Ling, and Nikias 1992].)

Next, we introduce the so-called *array transfer vector* (or *direction vector*):

$$a(\theta) = \left[H_1(\omega_c)e^{-i\omega_c\tau_1} \ldots H_m(\omega_c)e^{-i\omega_c\tau_m}\right]^T \tag{6.2.18}$$

Here, θ denotes the *source's direction of arrival*, which is the parameter of interest in our problem. Note that, since the transfer characteristics and positions of the sensors in the array are assumed to be known, the vector in (6.2.18) is a function of θ only, as indicated by notation (this fact will be illustrated shortly by means of a particular form of array). By making use of (6.2.18), we can write equation (6.2.16) as

$$y(t) = a(\theta)s(t) + e(t) \tag{6.2.19}$$

where

$$y(t) = [y_1(t) \ldots y_m(t)]^T$$
$$e(t) = [e_1(t) \ldots e_m(t)]^T$$

denote the *array's output vector* and the *additive noise vector*, respectively. It should be noted that θ enters in (6.2.18) not only through $\{\tau_k\}$ but also through $\{H_k(\omega_c)\}$. In some cases, the sensors may be considered to be *omnidirectional* over the DOA range of interest, and then the $\{H_k(\omega_c)\}_{k=1}^m$ are independent of θ. Sometimes, the sensors may also be assumed to be *identical*. Then, by *redefining the signal* ($H(\omega_c)s(t)$ is redefined as $s(t)$) and *selecting the first sensor as the reference point*, the expression (6.2.18) can be simplified to the following form:

$$a(\theta) = [1 \quad e^{-i\omega_c\tau_2} \dots e^{-i\omega_c\tau_m}]^T \tag{6.2.20}$$

The extension of equation (6.2.19) to the case of *multiple sources* is straightforward. Since the sensors in the array were assumed to be linear elements, a direct application of the *superposition principle* leads to the following *model of the array*:

$$\begin{aligned} y(t) &= [a(\theta_1)\dots a(\theta_n)]\begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix} + e(t) \triangleq As(t) + e(t) \\ \theta_k &= \text{the DOA of the } k\text{th source} \\ s_k(t) &= \text{the signal corresponding to the } k\text{th source} \end{aligned} \tag{6.2.21}$$

It is interesting to note that the previous model equation mainly relies on the *narrowband assumption* (6.2.13). The *planar wave* assumption made in the introductory part of this chapter has *not* been used so far. *This assumption is to be used when deriving the explicit dependence of* $\{\tau_k\}$ *as a function of* θ, as is illustrated next for an array with a special geometry.

Uniform Linear Array. Consider the array of m identical sensors uniformly spaced on a line, depicted in Figure 6.5. Such an array is commonly referred to as a uniform linear array (ULA). Let d denote the distance between two consecutive sensors, and let θ denote the DOA of the signal illuminating the array, as measured (counterclockwise) with respect to the normal to the line of sensors. Then, under the planar wave hypothesis and the assumption that the first sensor in the array is chosen as the reference point, we find that

$$\tau_k = (k-1)\frac{d\sin\theta}{c} \qquad \text{for } \theta \in [-90^\circ, 90^\circ] \tag{6.2.22}$$

where c is the propagation velocity of the impinging waveform (for example, the speed of light, in the case of electromagnetic waves). Inserting (6.2.22) into (6.2.20) gives

$$a(\theta) = \left[1, e^{-i\omega_c d\sin\theta/c}, \dots, e^{-i(m-1)\omega_c d\sin\theta/c}\right]^T \tag{6.2.23}$$

The restriction of θ to lie in the interval $[-90^\circ, 90^\circ]$ is a limitation of ULAs: two sources at locations symmetric with respect to the array line yield identical sets of delays $\{\tau_k\}$ and hence

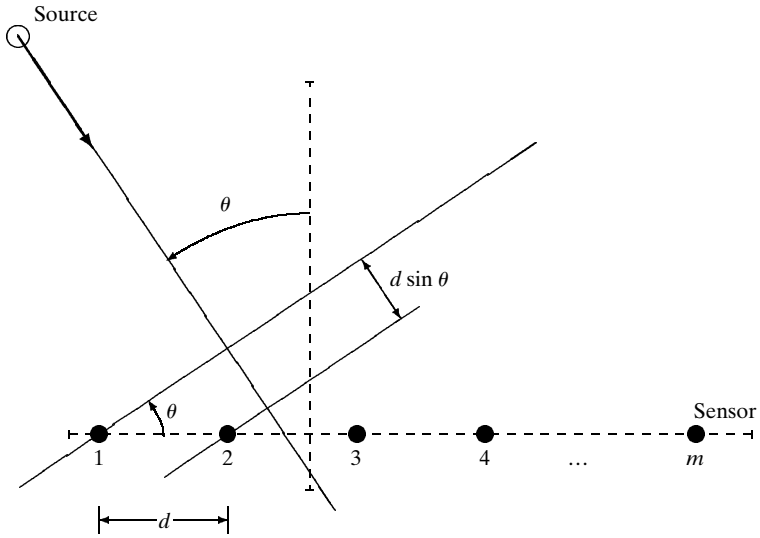


Figure 6.5 The uniform linear array scenario.

cannot be distinguished from one another. In practice, this ambiguity of ULAs is eliminated by using sensors that pass only signals whose DOAs are in $[-90^\circ, 90^\circ]$.

Let λ denote the *signal wavelength* (which is the distance traveled by the waveform in one period of the carrier):

$$\lambda = c/f_c, \qquad f_c = \omega_c/2\pi \tag{6.2.24}$$

Define

$$f_s = f_c \frac{d \sin\theta}{c} = \frac{d \sin\theta}{\lambda} \tag{6.2.25}$$

and

$$\boxed{\omega_s = 2\pi f_s = \omega_c \frac{d \sin\theta}{c}} \tag{6.2.26}$$

With this notation, the transfer vector (6.2.23) can be rewritten as

$$\boxed{a(\theta) = \begin{bmatrix} 1 & e^{-i\omega_s} & \dots & e^{-i(m-1)\omega_s} \end{bmatrix}^T} \tag{6.2.27}$$

This is a Vandermonde vector that is completely analogous to the vector made from the uniform samples of the sinusoidal signal $\{e^{-i\omega_s t}\}$. Let us explore this analogy a bit further.

First, by the previous analogy, ω_s is called the *spatial frequency*.

Second, if we were to sample a continuous-time sinusoidal signal with frequency ω_c, then, in order to avoid aliasing effects, the sampling frequency f_0 should satisfy (by the Nyquist sampling theorem)

$$f_0 > 2f_c \tag{6.2.28}$$

or, equivalently,

$$T_0 < \frac{T_c}{2} \tag{6.2.29}$$

where T_0 is the sampling period and T_c is the period of the continuous-time sinusoidal signal. Now, in the ULA case considered in this example, we see from (6.2.27) that the vector $a(\theta)$ is uniquely defined (i.e., there is no "spatial aliasing") if and only if ω_s is constrained as follows:

$$|\omega_s| < \pi \tag{6.2.30}$$

However, (6.2.30) is equivalent to

$$|f_s| < \frac{1}{2} \Longleftrightarrow d|\sin\theta| < \frac{\lambda}{2} \tag{6.2.31}$$

Note that this condition on d depends on θ. In particular, for a broadside source (i.e., a source with $\theta = 0^\circ$), (6.2.31) imposes *no* constraint on d. However, in general, we have no knowledge about the DOA of the source signal. Consequently, we would like (6.2.31) to hold for *any* θ, which leads to the following condition on d:

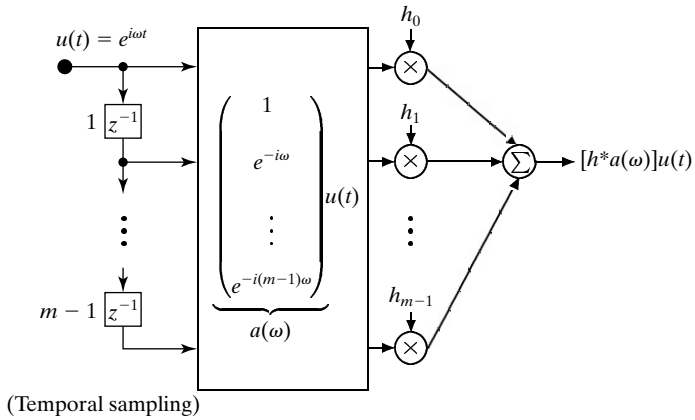
$$\boxed{d < \frac{\lambda}{2}} \tag{6.2.32}$$

We may think of the ULA as performing a uniform spatial sampling of the wavefield, so equation (6.2.32) simply says that the (spatial) sampling period d should be smaller than half of the signal wavelength. By analogy with (6.2.29), this result may be interpreted as a *spatial Nyquist sampling theorem*.

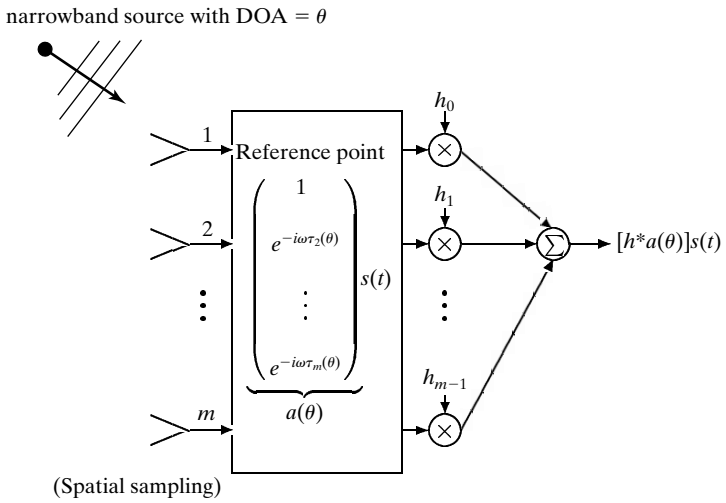
Equipped with the array model (6.2.21) derived previously, we can reduce the problem of DOA finding to that of estimating the parameters $\{\theta_k\}$ in (6.2.21). As there is a *direct analogy between (6.2.21) and the model (4.2.6) for sinusoidal signals in noise*, we may expect that most of the methods developed in Chapter 4 for (temporal) frequency estimation can also be used for DOA estimation. This is shown to be the case in the next sections, which briefly review the most important DOA finding methods.

6.3 NONPARAMETRIC METHODS

The methods to be described in this section *do not make any assumption about the covariance structure of the data*. As such, they may be considered to be "nonparametric." On the other hand, they assume that *the functional form of the array's transfer vector $a(\theta)$ is known*. Can we then still categorize them as "nonparametric methods"? The array performs a spatial sampling of the



(a) Temporal filter



(b) Spatial filter

Figure 6.6 Analogy between temporal sampling and filtering and the corresponding spatial operations performed by an array of sensors.

incoming wavefront, which is analogous to the temporal sampling done by the tapped-delay line implementation of a (temporal) finite impulse response (FIR) filter; see Figure 6.6. Thus, assuming that the form of $a(\theta)$ is available is no more restrictive than making the same assumption for $a(\omega)$ in Figure 6.6(a). In conclusion, the functional form of $a(\theta)$ characterizes the array as a *spatial sampling device* and, assuming it is known, should not be considered to be parametric (or model-based) information. As already mentioned, an array for which the functional form of $a(\theta)$ is known is said to be *calibrated*.

Figure 6.6 also makes an analogy between *temporal FIR filtering* and *spatial filtering* using an array of sensors. In what follows, we comment briefly on this analogy, since it is of interest for the nonparametric approach to DOA finding. In the time-series case, a FIR filter is defined by the relation

$$y_F(t) = \sum_{k=0}^{m-1} h_k u(t-k) \triangleq h^* y(t) \tag{6.3.1}$$

where $\{h_k\}$ are the filter weights, $u(t)$ is the input to the filter, and

$$h = [h_0 \dots h_{m-1}]^* \tag{6.3.2}$$

$$y(t) = [u(t) \dots u(t-m+1)]^T \tag{6.3.3}$$

Similarly, we can use the spatial samples $\{y_k(t)\}_{k=1}^m$ obtained with a sensor array to define a *spatial filter*:

$$\boxed{y_F(t) = h^* y(t)} \tag{6.3.4}$$

A temporal filter can be made to enhance or attenuate some selected frequency bands by choosing the vector h appropriately. More precisely, since the filter output for a sinusoidal input $u(t)$ is given by

$$y_F(t) = [h^* a(\omega)] u(t) \tag{6.3.5}$$

(where $a(\omega)$ is as defined, for instance, in Figure 6.6), then, by selecting h so that $h^* a(\omega)$ is large (small), we can enhance (attenuate) the power of $y_F(t)$ at frequency ω.

In direct analogy with (6.3.5), the (noise-free) spatially filtered output (as in (6.3.4)) of an array illuminated by a narrowband wavefront with complex envelope $s(t)$ and DOA equal to θ is given by (*cf.* (6.2.19)):

$$\boxed{y_F(t) = [h^* a(\theta)] s(t)} \tag{6.3.6}$$

This equation clearly shows that *the spatial filter can be selected to enhance (attenuate) the signals coming from a given direction* θ, by making $h^* a(\theta)$ in (6.3.6) large (small). This observation lies at the basis of the DOA-finding methods to be described in this section. All of these methods can be derived by using the *filter-bank approach* of Chapter 5. More specifically, assume that a filter h has been found such that

$$\boxed{\begin{array}{l}\text{(i) it passes undistorted the signals with a given DOA } \theta\text{; and} \\ \text{(ii) it attenuates all the other DOAs different from } \theta \text{ as much as possible.}\end{array}} \tag{6.3.7}$$

Then the power of the spatially filtered signal in (6.3.4),

$$E\left\{|y_F(t)|^2\right\} = h^*Rh, \qquad R = E\left\{y(t)y^*(t)\right\} \tag{6.3.8}$$

should give a good indication of the energy coming from direction θ. (Note that θ enters in (6.3.8) via h.) Hence, *h^*Rh should peak at the DOAs of the sources located in the array's viewing field* when evaluated over the DOA range of interest. This fact may be exploited for the purpose of DOA finding. Depending on the specific way in which the (loose) design objectives in (6.3.7) are formulated, the above approach can lead to different DOA estimation methods. In the following, we present *spatial extensions of the periodogram and Capon techniques*. The *RFB method* of Chapter 5 may also be extended to the spatial processing case, provided the array's geometry is such that the transfer vector $a(\theta + \alpha)$ can be factored as

$$a(\theta + \alpha) = D(\theta)a(\alpha) \tag{6.3.9}$$

where D is a unitary (possibly diagonal) matrix. Without such a property, the RFB spatial filter should be computed, *for each* θ, by solving an $m \times m$ eigendecomposition problem, which would be computationally prohibitive in most applications. It is not *a priori* obvious that an arbitrary array satisfies (6.3.9), so we do not consider the RFB approach in what follows.[1] Finally, we remark that a spatial filter satisfying the design objectives in (6.3.7) can be viewed as *forming a (reception) beam* in the direction θ, as pictorially indicated in Figure 6.7. Because of this interpretation, the methods resulting from this approach to the DOA-finding problem, in particular the method of the next subsection, are called *beamforming methods* [Van Veen and Buckley 1988; Johnson and Dudgeon 1992].

6.3.1 Beamforming

In view of (6.3.6), *condition* (i) of the filter-design problem (6.3.7) can be formulated as

$$h^*a(\theta) = 1 \tag{6.3.10}$$

In what follows, we assume that the transfer vector $a(\theta)$ has been normalized so that

$$a^*(\theta)a(\theta) = m \tag{6.3.11}$$

Note that, in the case of an array with identical sensors, the condition (6.3.11) is automatically met (*cf.* (6.2.20)).

Regarding *condition* (ii) in (6.3.7), if $y(t)$ in (6.3.8) were *spatially white with* $R = I$, then we would obtain the following expression for the power of the filtered signal:

$$E\left\{|y_F(t)|^2\right\} = h^*h \tag{6.3.12}$$

[1] Referring back to Chapter 5 may prove useful for understanding these comments on RFB and for several other discussions in this section.

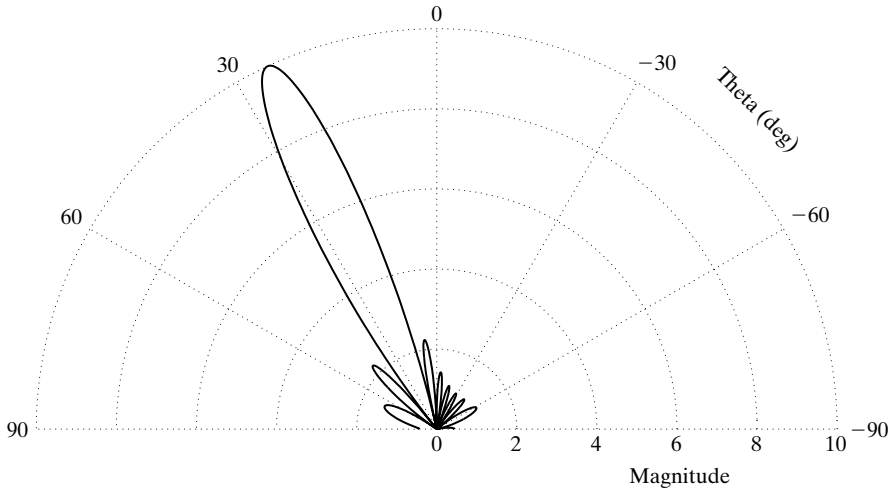


Figure 6.7 The response magnitude $|h^*a(\theta)|$, versus θ, of a spatial filter (or beamformer). Here, $h = a(\theta_0)$, where $\theta_0 = 25^\circ$ is the DOA of interest; the array is a 10-element ULA with $d = \lambda/2$.

This is different from zero for every θ (note that we cannot have $h = 0$, because of condition (6.3.10)), a fact which indicates that *a spatially white signal in the array output can be considered as having equal power for all directions* θ (in the same manner as a temporally white signal contains equal power in all frequency bands). We deduce from this observation that a natural mathematical formulation of condition (ii) would be to require that h minimize the power in (6.3.12). Hence, we are led to the following design problem:

$$\min_h h^*h \quad \text{subject to} \quad h^*a(\theta) = 1 \tag{6.3.13}$$

Because (6.3.13) is a special case of the optimization problem (5.4.7) in Chapter 5, we obtain the solution to (6.3.13) from (5.4.8) as

$$h = a(\theta)/a^*(\theta)a(\theta) \tag{6.3.14}$$

Making use of (6.3.11) reduces (6.3.14) to

$$h = a(\theta)/m \tag{6.3.15}$$

which, when inserted in (6.3.8), gives

$$E\left\{|y_F(t)|^2\right\} = a^*(\theta)Ra(\theta)/m^2 \tag{6.3.16}$$

The theoretical covariance matrix R in (6.3.16) cannot be (exactly) determined from the available finite sample $\{y(t)\}_{t=1}^{N}$; hence, it must be replaced by some estimate, such as

$$\hat{R} = \frac{1}{N}\sum_{t=1}^{N} y(t)y^*(t) \tag{6.3.17}$$

By doing so and omitting the factor $1/m^2$ in (6.3.16), which has no influence on the DOA estimates, we obtain the *beamforming method*, which estimates the DOAs as summarized in the next box.

The beamforming DOA estimates are given by the locations of the n highest peaks of the function

$$a^*(\theta)\hat{R}a(\theta) \tag{6.3.18}$$

When the estimated spatial spectrum in (6.3.18) is compared to the expression derived in Section 5.4 for the Blackman–Tukey periodogram, it is seen that *beamforming is a direct (spatial) extension of the periodogram*. In fact, the function in (6.3.18) may be thought of as being obtained by averaging the "spatial periodograms"

$$|a^*(\theta)y(t)|^2 \tag{6.3.19}$$

over the set of available "snapshots" ($t = 1, \ldots, N$).

The connection established in the previous paragraph, between beamforming and the (averaged) periodogram, suggests that the *resolution properties* of the beamforming method are analogous to those of the periodogram method. In fact, by an analysis similar to that in Chapters 2 and 5, it can be shown that the *beamwidth*[2] of the spatial filter used by beamforming is approximately equal to the inverse of the array's aperture (as measured in signal wavelengths). This sets a limit on the resolution achievable with beamforming (see Exercise 6.2):

Beamforming DOA resolution limit $\simeq$ wavelength / array "length" (6.3.20)

Next, we note that, as N increases, the sample spatial spectrum in (6.3.18) converges (under mild conditions) to (6.3.16), uniformly in θ. Hence, the beamforming estimates of the DOAs converge to the n maximum points of (6.3.16), as N tends to infinity. *If the array model (6.2.21) holds* (it has not been used so far!), *the noise $e(t)$ is spatially white and has the same power σ^2 in all sensors*, and *if there is only one source* (with DOA denoted by θ_0, for convenience), then R in (6.3.16) is given by

$$R = a(\theta_0)a^*(\theta_0)P + \sigma^2 I \tag{6.3.21}$$

[2]The beamwidth is the spatial counterpart of the temporal notion of bandwidth associated with a bandpass filter.

where $P = E\left\{|s(t)|^2\right\}$ denotes the signal power. Hence,

$$\begin{aligned} a^*(\theta)Ra(\theta) &= |a^*(\theta)a(\theta_0)|^2P + a^*(\theta)a(\theta)\sigma^2 \\ &\leq |a^*(\theta)a(\theta)||a^*(\theta_0)a(\theta_0)|P + \sigma^2 a^*(\theta)a(\theta) \\ &= m(mP + \sigma^2) \end{aligned} \tag{6.3.22}$$

where the inequality follows from the Cauchy–Schwartz lemma (see Result R22 in Appendix A) and the last equality follows from (6.3.11). The upper bound in (6.3.22) is achieved for $a(\theta) = a(\theta_0)$, which, under mild conditions, implies $\theta = \theta_0$. In conclusion, the *beamforming DOA estimate is consistent under the previous assumptions* ($n = 1$, *etc.*). *In the general case of multiple sources, however, the DOA estimates obtained with beamforming are inconsistent*. The (asymptotic) bias of these estimates may be significant if the sources are strongly correlated or closely spaced.

As was explained before, beamforming is the spatial analog of the Blackman–Tukey periodogram (with a certain covariance estimate) and of the Bartlett periodogram (if we interpret the m-dimensional snapshots in (6.3.19) as "subsamples" of the available "sample" $[y^T(1), \ldots, y^T(N)]^T$). Note, however, that the value of m in the periodogram methods can be chosen by the user, whereas, in the beamforming method, m is fixed. This difference might seem small at first, but it has a significant impact on the consistency properties of beamforming. More precisely, it can be shown that, for instance, the Bartlett periodogram estimates of *temporal frequencies* are *consistent* under the model (4.2.7), *provided that* m increases without bound as the number of samples N tends to infinity (e.g., we can set $m = N$, which yields the unmodified periodogram).[3] For beamforming, on the other hand, the value of m (i.e., the number of array elements) is *limited* by physical considerations. This prevents beamforming from providing consistent DOA estimates in the multiple-signal case. An additional difficulty is that, in the spatial scenario, the signals can be correlated with one another, whereas they are always uncorrelated in the temporal frequency estimation case. Explaining why this is so and completing a consistency analysis of the beamforming DOA estimates is left as an exercise for the reader.

Now, if the model (6.2.21) holds, if the minimum DOA separation is larger than the array beamwidth (which implies that m is sufficiently large), if the signals are uncorrelated, and if the noise is spatially white, then it is readily seen that the multiple-source spectrum (6.3.16) decouples (approximately) into n single-source spectra; this means that beamforming can provide reasonably accurate DOA estimates in such a case. In fact, in this case, beamforming can be shown to provide an approximation to the nonlinear LS DOA estimation method discussed in Section 6.4.1; see the remark in that section.

6.3.2 Capon Method

The derivation of the Capon method for array signal processing is entirely analogous to the derivation of the Capon method for time series data developed in Section 5.4 [Capon 1969; Lacoss

[3]The unmodified periodogram is an *inconsistent estimator* for *continuous* PSDs (as shown in Chapter 2). However, as already asserted, the plain periodogram estimates of *discrete (or line)* PSDs are *consistent*. Showing this is left as an exercise to the reader. (Make use of the covariance matrix model (4.2.7) with $m \to \infty$ and of the fact that the Fourier (or Vandermonde) vectors, at different frequencies, become orthogonal to one another as their dimension increases.)

1971]. The Capon spatial filter design problem is the following:

$$\min_h h^* R h \quad \text{subject to} \quad h^* a(\theta) = 1 \tag{6.3.23}$$

Hence, objective (i) in the general design problem (6.3.7) is ensured by constraining the filter exactly as in the beamforming approach. (See (6.3.10).) Objective (ii) in (6.3.7) is accomplished by requiring the filter to minimize the output power, when fed with the actual array data $\{y(t)\}$. Hence, in the Capon approach, objective (ii) is formulated in a "data-dependent" way, whereas it is formulated independently of the data in the beamforming method. As a consequence, the goal of the Capon filter steered to a certain direction θ is to attenuate any other signal that *actually impinges on the array* from a DOA $\neq \theta$, whereas the beamforming filter pays uniform attention to *all other* DOAs $\neq \theta$, even though there might be no incoming signal for many of those DOAs.

The solution to (6.3.23), as derived in Section 5.4, is given by

$$h = \frac{R^{-1} a(\theta)}{a^*(\theta) R^{-1} a(\theta)} \tag{6.3.24}$$

which, when inserted in the output power formula (6.3.8), leads to

$$E\left\{|y_F(t)|^2\right\} = \frac{1}{a^*(\theta) R^{-1} a(\theta)} \tag{6.3.25}$$

All that remains is to replace R in (6.3.25) by a sample estimate, such as $\hat{R}$ in (6.3.17), to obtain the Capon DOA estimator.

The Capon DOA estimates are obtained as the locations of the n largest peaks of the following function:

$$\frac{1}{a^*(\theta) \hat{R}^{-1} a(\theta)} \tag{6.3.26}$$

There is an implicit assumption in (6.3.26) that $\hat{R}^{-1}$ exists, but this can be ensured under weak conditions (in particular, $\hat{R}^{-1}$ exists with probability 1 if $N \geq m$ and if the noise term has a positive definite spatial covariance matrix). Note that the "spatial spectrum" in (6.3.26) corresponds to the "CM-Version 1" PSD for time series. (See equation (5.4.12) in Section 5.4.) A Capon spatial spectrum similar to the "CM-Version 2" PSD formula (see (5.4.17)) might also be derived, but it appears to be more complicated than the time series formula if the array is not a ULA.

Capon DOA estimation has been found empirically to possess superior performance as compared with beamforming. The common advantage of these two nonparametric methods is that they do not assume anything about the statistical properties of the data, and, therefore, they can be used in situations where we lack information about these properties. On the other hand,

in the cases where such information is available, for example in the form of a covariance model of the data, a nonparametric approach does not give the performance that one can achieve with a parametric (model-based) approach. The parametric approach to DOA estimation is the subject of the next section.

6.4 PARAMETRIC METHODS

In this section, *we postulate the array model (6.2.21)*. Furthermore, the noise $e(t)$ is assumed to be spatially white with components having identical variance:

$$E\left\{e(t)e^*(t)\right\} = \sigma^2 I \tag{6.4.1}$$

In addition, the signal covariance matrix

$$P = E\left\{s(t)s^*(t)\right\} \tag{6.4.2}$$

is assumed to be *nonsingular* (but not necessarily diagonal; hence, the signals may be (partially) correlated). When the signals are fully correlated, so that P is singular, they are said to be *coherent*. Finally, we assume that the signals and the noise are uncorrelated with one another.

Under the previous assumptions, the theoretical covariance matrix of the array output vector is given by

$$R = E\left\{y(t)y^*(t)\right\} = APA^* + \sigma^2 I \tag{6.4.3}$$

There is a direct analogy between the array models (6.2.21) and (6.4.3) and the corresponding models encountered in our discussion of the sinusoids-in-noise case in Chapter 4. More specifically, the "nonlinear regression" model (6.2.21) of the array is analogous to (4.2.6), and the array covariance model (6.4.3) is much the same as (4.2.7). The consequence of these analogies is that *all methods introduced in Chapter 4 for frequency estimation can also be used for DOA estimation* without any essential modification. In the following, we briefly review these methods, with a view to pointing out any differences from the frequency-estimation application. When the assumed array model is a good representation of reality, the parametric DOA estimation methods provide highly accurate DOA estimates, even in adverse situations (such as low SNR scenarios). Our main thrust in this text has been to understand the basic ideas behind the presented spectral estimation methodologies, so we do not dwell on the details of the analysis required to establish the statistical properties of the DOA estimators discussed in what follows; see, however, Appendix B for a discussion on the Cramér–Rao bound and the best accuracy achievable in DOA estimation problems. Such analysis details are available in [Stoica and Nehorai 1989a; Stoica and Nehorai 1990; Stoica and Sharman 1990; Stoica and Nehorai 1991; Viberg and Ottersten 1991; Rao and Hari 1993]. For reviews of many of the recent advances in spatial-spectral analysis, the reader can consult [Pillai 1989], [Ottersten, Viberg, Stoica, and Nehorai 1993], and [Van Trees 2002].

6.4.1 Nonlinear Least-Squares Method

This method finds the unknown DOAs as the minimizing elements of the following function:

$$f = \frac{1}{N}\sum_{t=1}^{N} \| \, y(t) - As(t) \, \|^2 \tag{6.4.4}$$

Minimization with respect to $\{s(t)\}$ (see Result R32 in Appendix A) gives

$$s(t) = (A^*A)^{-1}A^*y(t) \qquad t = 1, \ldots, N \tag{6.4.5}$$

By inserting (6.4.5) into (6.4.4), we get the following concentrated nonlinear least-squares (LS) criterion:

$$\begin{aligned} f &= \frac{1}{N}\sum_{t=1}^{N} \| \, \{I - A(A^*A)^{-1}A^*\}y(t) \, \|^2 \\ &= \frac{1}{N}\sum_{t=1}^{N} y^*(t)\left[I - A(A^*A)^{-1}A^*\right]y(t) \\ &= \mathrm{tr}\{[I - A(A^*A)^{-1}A^*]\hat{R}\} \end{aligned} \tag{6.4.6}$$

The second equality in (6.4.6) follows from the fact that the matrix $I - A(A^*A)^{-1}A^*$ is idempotent (it is the orthogonal projector onto $\mathcal{N}(A^*)$) and the third equality follows from the properties of the trace operator. (See Result R8 in Appendix A.) From (6.4.6), the nonlinear LS DOA estimates are given by

$$\{\hat{\theta}_k\} = \arg\max_{\{\theta_k\}} \mathrm{tr}\left[A(A^*A)^{-1}A^*\hat{R}\right] \tag{6.4.7}$$

Remark: Much as in the frequency-estimation case, it can be shown that beamforming provides an approximate solution to the previous nonlinear LS problem whenever the DOAs are known to be well separated. To see this, let us assume that we restrict the search for the maximizers of (6.4.7) to a set of well-separated DOAs (according to the *a priori* information that the true DOAs belong to this set). In such a set, $A^*A \simeq mI$ under weak conditions; hence, the function in (6.4.7) can be approximately written as

$$\mathrm{tr}\left[A(A^*A)^{-1}A^*\hat{R}\right] \simeq \frac{1}{m}\sum_{k=1}^{n} a^*(\theta_k)\hat{R}a(\theta_k)$$

Paralleling the discussion following equation (4.3.16) in Chapter 4, we can show that the beamforming DOA estimates maximize the right-hand side of the previous equation over the set under

consideration. With this observation, the proof of the fact that the computationally efficient beamforming method provides an approximate solution to (6.4.7) in scenarios with well-separated DOAs is concluded. ■

One difference between (6.4.7) and the corresponding optimization problem in the frequency-estimation application (see (4.3.8) in Section 4.3) lies in the fact that, in the frequency-estimation application, only one "snapshot" of data is available, in contrast to the N snapshots available in the DOA-estimation application. Another, more important difference is that, for non-ULA cases, the matrix A in (6.4.7) does not have the Vandermonde structure of the corresponding matrix in (4.3.8). As a consequence, several of the algorithms used to (approximately) solve the frequency-estimation problem (such as the one in [KUMARESAN, SCHARF, AND SHAW 1986] and [BRESLER AND MACOVSKI 1986]) are no longer applicable to solving (6.4.7) unless the array is a ULA.

6.4.2 Yule–Walker Method

The matrix Γ, which lies at the basis of the Yule–Walker method (see Section 4.4), can be constructed from any block of R in (6.4.3) that does not include diagonal elements. To be more precise, partition the array model (6.2.21) into the following two nonoverlapping parts:

$$y(t) = \begin{bmatrix} \bar{y}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} \bar{A} \\ \tilde{A} \end{bmatrix} s(t) + \begin{bmatrix} \bar{e}(t) \\ \tilde{e}(t) \end{bmatrix} \tag{6.4.8}$$

Because $\bar{e}(t)$ *and* $\tilde{e}(t)$ *are uncorrelated* (by assumption), we have

$$\boxed{\Gamma \triangleq E\left\{\bar{y}(t)\tilde{y}^*(t)\right\} = \bar{A}P\tilde{A}^*} \tag{6.4.9}$$

which is assumed to be of dimension $M \times L$ (with $M + L = m$). For

$$\boxed{M > n, \qquad L > n} \tag{6.4.10}$$

(which cannot hold unless $m > 2n$), the rank of Γ is equal to n (under weak conditions) and the $(L-n)$-dimensional null space of this matrix contains complete information about the DOAs. To see this, let G be an $L \times (L-n)$ matrix whose columns form a basis of $\mathcal{N}(\Gamma)$. (G can be obtained from the SVD of Γ; see Result R15 in Appendix A.) Then we have $\Gamma G = 0$, which implies (using the fact that $\text{rank}(\bar{A}P) = n$) that

$$\tilde{A}^* G = 0$$

This observation can be used, in the manner of Sections 4.4 (YW) and 4.5 (MUSIC), to estimate the DOAs from a sample estimate of Γ, such as

$$\hat{\Gamma} = \frac{1}{N} \sum_{t=1}^{N} \bar{y}(t)\tilde{y}^*(t) \tag{6.4.11}$$

Unlike all the other methods discussed in what follows, *the Yule–Walker method does not impose the rather stringent condition (6.4.1)*. The Yule–Walker method requires only that $E\{\bar{e}(t)\tilde{e}^*(t)\} = 0$, which is a much weaker assumption. This is a distinct advantage of the Yule–Walker method (see [Viberg, Stoica, and Ottersten 1995] for details). Its relative drawback is that it cannot be used unless $m > 2n$ (all the other methods require only that $m > n$); in general, it has been found to provide accurate DOA estimates only in those applications involving large-aperture arrays.

Interestingly enough, whenever the condition (6.4.1) holds (i.e., the noise at the array output is spatially white), we can use a modification of the technique above that does not require that $m > 2n$ [Fuchs 1996]. To see this, let

$$\tilde{\Gamma} \triangleq E\left\{y(t)\tilde{y}^*(t)\right\} = R\begin{bmatrix} 0 \\ I_L \end{bmatrix} \qquad (m \times L)$$

where $\tilde{y}(t)$ is as defined in (6.4.8); hence $\tilde{\Gamma}$ is made from the last L columns of R. By making use of the expression (6.4.3) for R, we obtain

$$\tilde{\Gamma} = AP\tilde{A}^* + \sigma^2\begin{bmatrix} 0 \\ I_L \end{bmatrix} \tag{6.4.12}$$

Because the noise terms in $y(t)$ and $\tilde{y}(t)$ are correlated, the noise is still present in $\tilde{\Gamma}$ (as can be seen from (6.4.12)); hence, $\tilde{\Gamma}$ is not really a YW matrix. Nevertheless, $\tilde{\Gamma}$ has a property similar to that of the YW matrix Γ, as we now show.

First, observe that

$$\tilde{\Gamma}^*\tilde{\Gamma} = \tilde{A}(2\sigma^2 P + PA^*AP)\tilde{A}^* + \sigma^4 I$$

The matrix $2\sigma^2 P + PA^*AP$ is readily shown to be nonsingular if and only if P is nonsingular. As $\tilde{\Gamma}^*\tilde{\Gamma}$ has the same form as R in (6.4.3), we conclude that (for $m \geq L > n$) the $L \times (L-n)$ matrix $\tilde{G}$, whose columns are the eigenvectors of $\tilde{\Gamma}^*\tilde{\Gamma}$ that correspond to the multiple minimum eigenvalue of σ^4, satisfies

$$\tilde{A}^*\tilde{G} = 0 \tag{6.4.13}$$

The columns of $\tilde{G}$ are also equal to the $(L-n)$ right singular vectors of $\tilde{\Gamma}$ corresponding to the multiple minimum singular value of σ^2. For numerical precision reasons, $\tilde{G}$ should be computed from the singular vectors of $\tilde{\Gamma}$ rather than from the eigenvectors of $\tilde{\Gamma}^*\tilde{\Gamma}$ (see Section A.8.2).

Because (6.4.13) has the same form as $\tilde{A}^*G = 0$, we can use (6.4.13) for subspace-based DOA estimation in exactly the same way as we used $\tilde{A}^*G = 0$ (see equation (4.5.6) and the discussion following it in Chapter 4). Note that, for the method based on $\tilde{\Gamma}$ to be usable, we require only that

$$m \geq L > n \tag{6.4.14}$$

instead of the more restrictive conditions $\{m - L > n,\ L > n\}$ (see (6.4.10)) required in the YW method based on Γ. Observe that (6.4.14) can always be satisfied if $m > n$, whereas (6.4.10) requires that $m > 2n$. Finally, note that Γ is made from the first $m - L$ rows of $\tilde{\Gamma}$

and hence contains "less information" than $\tilde{\Gamma}$; this provides a quick intuitive explanation of why the method based on Γ requires more sensors to be applicable than does the method based on $\tilde{\Gamma}$.

6.4.3 Pisarenko and MUSIC Methods

The MUSIC algorithm (with Pisarenko as a special case), developed in Section 4.5 for the frequency-estimation application, can be used without modification for DOA estimation [Bienvenu 1979; Schmidt 1979; Barabell 1983]. There are only minor differences between the DOA and the frequency-estimation applications of MUSIC, as pointed out next.

First, in the spatial application, we can choose between the Spectral and Root MUSIC estimators only in the case of a ULA. For most of the other array geometries, *only Spectral MUSIC is applicable*.

Second, *the standard MUSIC algorithm (4.5.15) breaks down in the case of coherent signals*, because, in that case, the rank condition (4.5.1) no longer holds. (Such a situation cannot occur in the frequency-estimation application, because P is always (diagonal and) nonsingular there.) However, the *modified MUSIC algorithm (outlined at the end of Section 4.5) can be used when the signals are coherent, provided that the array is uniform and linear*. This is so because the property (4.5.23), on which the modified MUSIC algorithm is based, continues to hold even if P is singular. (See Exercise 6.14.)

6.4.4 Min–Norm Method

There is no essential difference between the use of the Min–Norm method for frequency estimation and for DOA-finding in the noncoherent case. As for MUSIC, in the DOA-estimation application, the Min–Norm method should not be used in scenarios with coherent signals, and the Root Min–Norm algorithm can only be used in the ULA case [Kumaresan and Tufts 1983]. In addition, the key property that the true DOAs are asymptotically the *unique* solutions of the Min–Norm estimation problem holds in the ULA case (see Complement 6.5.1), but not necessarily for other array geometries.

6.4.5 ESPRIT Method

In the ULA case, ESPRIT can be used for DOA estimation exactly as it is for frequency estimation. (See Section 4.7.) In the non-ULA case, ESPRIT can be used only in certain situations. More precisely, and unlike the other algorithms in this section, ESPRIT can be used for DOA finding only if *the array contains two identical subarrays that are displaced by a known displacement vector* [Roy and Kailath 1989; Stoica and Nehorai 1991]. Mathematically, this condition can be formulated as follows: Let $\bar{m}$ denote the number of sensors in the two twin subarrays, and let A_1 and A_2 denote the submatrices of A corresponding to these subarrays. The sensors in the array are numbered arbitrarily, so it constitutes no restriction to assume that A_1 is made from the first $\bar{m}$ rows in A and A_2 from the last $\bar{m}$:

$$A_1 = [I_{\bar{m}} \quad 0]A \qquad (\bar{m} \times n) \tag{6.4.15}$$

$$A_2 = [0 \quad I_{\bar{m}}]A \qquad (\bar{m} \times n) \tag{6.4.16}$$

Here $I_{\bar{m}}$ denotes the $\bar{m} \times \bar{m}$ identity matrix. Note that the two subarrays overlap if $\bar{m} > m/2$; otherwise, they might not overlap. If the array is purposely built to meet ESPRIT's subarray condition, then, normally, $\bar{m} = m/2$, and the two subarrays are nonoverlapping.

Mathematically, the ESPRIT requirement means that

$$A_2 = A_1 D \tag{6.4.17}$$

where

$$D = \begin{bmatrix} e^{-i\omega_c \tau(\theta_1)} & & 0 \\ & \ddots & \\ 0 & & e^{-i\omega_c \tau(\theta_n)} \end{bmatrix} \tag{6.4.18}$$

and where $\tau(\theta)$ denotes the time needed by a wavefront impinging upon the array from the direction θ to travel between (the "reference points" of) the two twin subarrays. If the angle of arrival θ is measured with respect to the perpendicular of the line between the subarrays' center points, then a calculation similar to the one that led to (6.2.22) shows that

$$\tau(\theta) = d \sin(\theta)/c \tag{6.4.19}$$

where d is the distance between the two subarrays. Hence, estimates of the DOAs can readily be derived from estimates of the diagonal elements of D in (6.4.18).

Equations (6.4.17) and (6.4.18) are basically equivalent to (4.7.3) and (4.7.4) in Section 4.7; hence, the ESPRIT DOA estimation method is analogous to the ESPRIT frequency estimator.

The ESPRIT DOA estimation method, like the ESPRIT frequency estimator, computes the DOA estimates by solving an $n \times n$ eigenvalue problem. *There is no search involved*, in contrast to the previous methods; in addition, *there is no problem of separating the "signal DOAs" from the "noise DOAs,"* once again in contrast to the Yule–Walker, MUSIC, and Min–Norm methods. However, unlike these other methods, *ESPRIT can only be used with the special array configuration described earlier*. In particular, this requirement limits the number of resolvable sources at $n < \bar{m}$ (as both A_1 and A_2 must have full column rank). Note that *the two subarrays do not need to be calibrated*, although they need to be identical, and *ESPRIT can be sensitive to differences between the two subarrays* in the same way that Yule–Walker, MUSIC, and Min–Norm are sensitive to imperfections in array calibration. Finally, note that, like the other DOA-finding algorithms presented in this section (with the exception of the NLS method), ESPRIT is not usable in the case of coherent signals.

6.5 COMPLEMENTS

6.5.1 On the Minimum–Norm Constraint

As explained in Section 6.4.4, the Root Min–Norm (temporal) frequency estimator, introduced in Section 4.6, can be used without modification for DOA estimation with a uniform linear array. Using the definitions and notation in Section 4.6, let $\hat{g} = [1\ \hat{g}_1 \ldots \hat{g}_{m-1}]^T$ denote the vector in

$\mathcal{R}(\hat{G})$ that has the first element equal to one and minimum Euclidean norm. Then, the Root Min–Norm DOA estimates are obtained from the roots of the polynomial

$$\hat{g}(z) = 1 + \hat{g}_1 z^{-1} + \cdots + \hat{g}_{m-1} z^{-(m-1)} \tag{6.5.1}$$

that are located nearest the unit circle. (See the description of Min–Norm in Section 4.6.) As N increases, the polynomial in (6.5.1) approaches

$$g(z) = 1 + g_1 z^{-1} + \cdots + g_{m-1} z^{-(m-1)} \tag{6.5.2}$$

where $g = [1\ g_1 \ldots g_{m-1}]^T$ is the minimum–norm vector in $\mathcal{R}(G)$. In this complement, we show that (6.5.2) has n zeroes at $\{e^{-i\omega_k}\}_{k=1}^n$ (the so-called "signal zeroes") and $(m-n-1)$ extraneous zeroes situated *strictly inside* the unit circle (the latter are normally called "noise zeroes"); here $\{\omega_k\}_{k=1}^n$ either are temporal frequencies or are spatial frequencies (as in (6.2.27)).

Let $g = [1, g_1, \ldots, g_{m-1}]^T \in \mathcal{R}(G)$. Then (4.2.4) and (4.5.6) imply that

$$a^*(\omega_k) \begin{bmatrix} 1 \\ g_1 \\ \vdots \\ g_{m-1} \end{bmatrix} = 0 \qquad \Longleftrightarrow$$

$$1 + g_1 e^{i\omega_k} + \cdots + g_{m-1} e^{i(m-1)\omega_k} = 0 \quad (\text{for } k = 1, \ldots, n) \tag{6.5.3}$$

Hence, any polynomial $g(z)$ whose coefficient vector belongs to $\mathcal{R}(G)$ must have zeroes at $\{e^{-i\omega_k}\}_{k=1}^n$, and thus it can be factored as

$$g(z) = g_s(z) g_n(z) \tag{6.5.4}$$

where

$$g_s(z) = \prod_{k=1}^{n} (1 - e^{-i\omega_k} z^{-1})$$

The $(m-n-1)$-degree polynomial $g_n(z)$ in (6.5.4) contains the noise zeroes, and at this point is arbitrary. (As the coefficients of $g_n(z)$ vary, the vectors made from the corresponding coefficients of $g(z)$ span $\mathcal{R}(G)$.)

Next, assume that g satisfies the minimum–norm constraint:

$$\sum_{k=0}^{m-1} |g_k|^2 = \min \qquad (g_0 \triangleq 1) \tag{6.5.5}$$

By using Parseval's theorem (see (1.2.6)), we can rewrite (6.5.5) as

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |g(\omega)|^2 \, d\omega = \min \Longleftrightarrow \frac{1}{2\pi} \int_{-\pi}^{\pi} |g_n(\omega)|^2 \, |g_s(\omega)|^2 \, d\omega = \min \tag{6.5.6}$$

(where, by convention, $g(\omega) = g(z)\big|_{z=e^{i\omega}}$). Since $g_s(z)$ in (6.5.4) is fixed, the minimization in (6.5.6) is over $g_n(z)$.

To proceed, some additional notation is required. Let

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

and let $y(t)$ be a signal whose PSD is equal to $|g_s(\omega)|^2$; hence, $y(t)$ is an nth-order MA process. By making use of (1.3.9) and (1.4.9), along with the previous notation, we can write (6.5.6) in the following equivalent form:

$$\min_{\{\alpha_k\}} E\left\{|y(t) + \alpha_1 y(t-1) + \cdots + \alpha_{m-n-1} y(t-m+n+1)|^2\right\} \tag{6.5.7}$$

The minimizing coefficients $\{\alpha_k\}$ are given by the solution to a Yule–Walker system of equations similar to (3.4.6). (To show this, parallel the calculation leading to (3.4.8) and (3.4.12).) Since the covariance matrix, of any finite dimension, associated with a moving-average signal is positive definite, it follows that

- the coefficients $\{\alpha_k\}$, and hence $\{g_k\}$, are *uniquely* determined by the minimum–norm constraint
- the polynomial $g_n(z)$ whose coefficients are obtained from (6.5.7) has all its zeroes *strictly inside* the unit circle (*cf.* Exercise 3.8)

which was to be proven.

Thus, the choice of $\hat{g}$ in the Min–Norm algorithm makes it possible to separate the signal zeroes from the noise zeroes, at least for data samples that are sufficiently long. (For small or medium-sized samples, it might happen that noise zeroes get closer to the unit circle than signal zeroes, which would lead to spurious frequency or DOA estimates.)

As a final remark, note, from (6.5.6), that there is little reason for $g_n(z)$ to have zeroes in the sectors where the signal zeroes are present (since the integrand in (6.5.6) is already quite small for ω values close to $\{\omega_k\}_{k=1}^n$). Hence, we can expect the extraneous zeroes to be more or less uniformly distributed inside the unit circle, in sectors that do not contain signal zeroes (see, e.g., [Kumaresan 1983].)

For more details on the topic of this complement, see [Tufts and Kumaresan 1982; Kumaresan 1983].

6.5.2 NLS Direction-of-Arrival Estimation for a Constant-Modulus Signal

The NLS estimation of the DOA of a *single signal* impinging on an array of sensors is obtained by minimizing the criterion (6.4.4) with $n = 1$,

$$\sum_{t=1}^{N} \|y(t) - a(\theta)s(t)\|^2 \tag{6.5.8}$$

with respect to $\{s(t)\}_{t=1}^{N}$ and θ. The result is obtained from equation (6.4.7), which for $n = 1$ reduces to

$$\hat{\theta} = \arg\max_{\theta} a^*(\theta)\hat{R}a(\theta) = \arg\max_{\theta} \sum_{t=1}^{N} \left|a^*(\theta)y(t)\right|^2 \tag{6.5.9}$$

This, of course, is nothing but the *beamforming DOA estimate* for $n = 1$; see (6.3.18). Hence, as expected (see the Remark following (6.4.7) and also (4.3.11)), the NLS estimate of the DOA of an *arbitrary* signal coincides with the beamforming estimate.

In this complement, we will solve the NLS direction-of-arrival estimation problem in (6.5.8), under the assumption that $\{s(t)\}$ is a *constant-modulus signal*; that is,

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

where $\alpha > 0$ denotes the unknown signal amplitude and $\{\phi(t)\}$ is its unknown phase sequence. We assume $\alpha > 0$ to avoid a phase ambiguity in $\{\phi(t)\}$. Signals of this type are often encountered in communication applications with phase-modulated waveforms.

Inserting (6.5.10) in (6.5.8) yields the following criterion that is to be minimized with respect to $\{\phi(t)\}_{t=1}^{N}$, α, and θ:

$$\begin{aligned}
&\sum_{t=1}^{N} \left\|y(t) - \alpha e^{i\phi(t)}a(\theta)\right\|^2 \\
&\quad = \sum_{t=1}^{N} \left\{\|y(t)\|^2 + \alpha^2\|a(\theta)\|^2 - 2\alpha \operatorname{Re}\left[a^*(\theta)y(t)e^{-i\phi(t)}\right]\right\}
\end{aligned} \tag{6.5.11}$$

It follows from (6.5.11) that the NLS estimate of $\{\phi(t)\}_{t=1}^{N}$ is given by the maximizer of the function

$$\begin{aligned}
\operatorname{Re}\left[a^*(\theta)y(t)e^{-i\phi(t)}\right] &= \operatorname{Re}\left[\left|a^*(\theta)y(t)\right| e^{i\arg[a^*(\theta)y(t)]}e^{-i\phi(t)}\right] \\
&= \left|a^*(\theta)y(t)\right| \cos\left[\arg\big(a^*(\theta)y(t)\big) - \phi(t)\right]
\end{aligned} \tag{6.5.12}$$

which is easily seen to be

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

From (6.5.11)–(6.5.13), along with the assumption that $\|a(\theta)\|$ is constant (which is also used to derive (6.5.9)), we can readily verify that the NLS estimate of θ for the constant modulus signal case is given by

$$\boxed{\hat{\theta} = \arg\max_{\theta} \sum_{t=1}^{N} \left|a^*(\theta)y(t)\right|} \tag{6.5.14}$$

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

$$\hat{\alpha} = \frac{1}{N\|a(\hat{\theta})\|^2} \sum_{t=1}^{N} \left| a^*(\hat{\theta}) y(t) \right| \tag{6.5.15}$$

Remark: It follows easily from the preceding derivation that, if α is known (as could be the case when the emitted signal has a known amplitude that is not significantly distorted during propagation), the NLS estimates of θ and $\{\phi(t)\}$ are still given by (6.5.13) and (6.5.14). ■

Interestingly, the only difference between the beamformer for an arbitrary signal, (6.5.9), and the beamformer for a constant-modulus signal, (6.5.14), is that *the "squaring operation" is missing in the latter*. This difference is somewhat analogous to the one pointed out in Complement 4.9.4, even though the models considered there and in this complement are rather different from one another.

For more details on the subject of this complement, see [Stoica and Besson 2000] and its references.

6.5.3 Capon Method: Further Insights and Derivations

The spatial filter (or beamformer) used in the beamforming method is independent of data. In contrast, the Capon spatial filter is data dependent, or *data adaptive*; see equation (6.3.24). It is this data adaptivity that confers on the Capon method better resolution and significantly reduced leakage as compared with the beamforming method.

An interesting fact about the Capon method for temporal or spatial spectral analysis is that it can be derived in several ways. The standard derivation is given in Section 6.3.2. This complement presents four additional derivations of the Capon method, each not as well known as the standard derivation. Each of the derivations presented here is based on an intuitively appealing design criterion. Collectively, they provide further insights into the features and possible interpretations of the Capon method.

APES-Like Derivation

Let θ denote a generic DOA, and consider equation (6.2.19),

$$y(t) = a(\theta)s(t) + e(t) \tag{6.5.16}$$

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

$$\min_{h,\sigma_s^2} E\left\{ |h^* y(t) - s(t)|^2 \right\} \tag{6.5.17}$$

Of course, the signal $s(t)$ in (6.5.17) is not known. However, as we show shortly, (6.5.17) does not depend on $s(t)$ but only on its power σ_s^2, so the fact that $s(t)$ in (6.5.17) is unknown does not pose a problem. Also, note that the vector h in (6.5.17) is *not* constrained, as it is in (6.3.24).

Assuming that $s(t)$ in (6.5.16) is uncorrelated with the noise-plus-interference term $e(t)$, we obtain

$$E\left\{y(t)s^*(t)\right\} = a(\theta)\sigma_s^2 \tag{6.5.18}$$

which implies that

$$\begin{aligned} E\left\{|h^*y(t) - s(t)|^2\right\} &= h^*Rh - h^*a(\theta)\sigma_s^2 - a^*(\theta)h\sigma_s^2 + \sigma_s^2 \\ &= \left[h - \sigma_s^2 R^{-1}a(\theta)\right]^* R\left[h - \sigma_s^2 R^{-1}a(\theta)\right] \\ &\quad + \sigma_s^2\left[1 - \sigma_s^2 a^*(\theta)R^{-1}a(\theta)\right] \end{aligned} \tag{6.5.19}$$

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

$$h = \frac{R^{-1}a(\theta)}{a^*(\theta)R^{-1}a(\theta)} \tag{6.5.20}$$

$$\sigma_s^2 = \frac{1}{a^*(\theta)R^{-1}a(\theta)} \tag{6.5.21}$$

which coincides with the Capon solution in (6.3.24) and (6.3.25). To obtain σ_s^2 in (6.5.21), we used the fact that the criterion in (6.5.19) should be greater than or equal to zero for any h and σ_s^2.

The LS-fitting criterion in (6.5.17) is *reminiscent of the APES approach* discussed in Complement 5.6.4. The use of APES for array processing is discussed in Complement 6.5.6, under the assumption that $\{s(t)\}$ is an unknown *deterministic* sequence. Interestingly, using the APES design principle in the above manner, under the assumption that the signal $s(t)$ in (6.5.16) is *stochastic*, leads to the Capon method.

Inverse-Covariance-Fitting Derivation

The covariance matrix of the signal term $a(\theta)s(t)$ in (6.5.16) is given by

$$\sigma_s^2 a(\theta)a^*(\theta) \tag{6.5.22}$$

We can obtain the beamforming method (see Section 6.3.1) by fitting (6.5.22) to R in a least-squares sense:

$$\begin{aligned} &\min_{\sigma_s^2} \|R - \sigma_s^2 a(\theta)a^*(\theta)\|^2 \\ &\quad = \min_{\sigma_s^2}\{\text{constant} + \sigma_s^4[a^*(\theta)a(\theta)]^2 - 2\sigma_s^2 a^*(\theta)Ra(\theta)\} \end{aligned} \tag{6.5.23}$$

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

$$\sigma_s^2 = \frac{1}{m^2} a^*(\theta) R a(\theta) \tag{6.5.24}$$

which coincides with the beamforming estimate of the power coming from DOA θ (see (6.3.16)).

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

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

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

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

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

$$\sigma_s^2 a(\theta)a^*(\theta) = \left[\sigma_s^2 \|a(\theta)\|^2\right] \left[\frac{a(\theta)}{\|a(\theta)\|}\right] \left[\frac{a(\theta)}{\|a(\theta)\|}\right]^* \triangleq \sigma u v^* \tag{6.5.27}$$

is the singular-value decomposition (SVD) of $\sigma_s^2 a(\theta)a^*(\theta)$. Inserting (6.5.26) into (6.5.25) leads to the problem

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

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

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

It is worth noting that in the present *covariance-fitting-based derivation*, the signal power σ_s^2 is estimated directly *without* the need to first obtain an intermediate spatial filter h. The remaining two derivations of the Capon method are of the same type.

Weighted-Covariance-Fitting Derivation

The least-squares criterion in (6.5.23), which yields the beamforming method, does not take into account the fact that the sample estimates of the different elements of the data covariance matrix do not have the same accuracy. It was shown (e.g., in [Ottersten, Stoica, and Roy 1998] and its references) that the following *weighted LS covariance-fitting criterion* takes the accuracies of

the different elements of the sample covariance matrix into account in an *optimal manner*:

$$\min_{\sigma_s^2} \left\| R^{-1/2} \left[R - \sigma_s^2 a(\theta) a^*(\theta) \right] R^{-1/2} \right\|^2 \tag{6.5.30}$$

Here, $R^{-1/2}$ denotes the Hermitian square root of R^{-1}. By a straightforward calculation, we can rewrite the criterion in (6.5.30) in the following equivalent form:

$$\begin{aligned} &\left\| I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2} \right\|^2 \\ &\quad = \text{constant} - 2\sigma_s^2 a^*(\theta) R^{-1} a(\theta) + \sigma_s^4 \left[a^*(\theta) R^{-1} a(\theta) \right]^2 \end{aligned} \tag{6.5.31}$$

The minimization of (6.5.31) with respect to σ_s^2 yields

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

which coincides with the Capon solution in (6.3.26).

Constrained-Covariance-Fitting Derivation

The final derivation of the Capon method that we will present is also based on a covariance-fitting criterion, but in a manner quite different from those in the previous two derivations. Our goal here is still to obtain the signal power by fitting $\sigma_s^2 a(\theta) a^*(\theta)$ to R, but now we explicitly impose the condition that the residual covariance matrix, $R - \sigma_s^2 a(\theta) a^*(\theta)$, should be positive semidefinite, and we "minimize" the approximation (or fitting) error by choosing the maximum possible value of σ_s^2 for which this condition holds. Mathematically, σ_s^2 is the solution to the following constrained covariance-fitting problem:

$$\max_{\sigma_s^2} \sigma_s^2 \quad \text{subject to } R - \sigma_s^2 a(\theta) a^*(\theta) \geq 0 \tag{6.5.32}$$

The solution to (6.5.32) can be obtained in the following way, which is a simplified version of the original derivation in [Marzetta 1983]: Let $R^{-1/2}$ again denote the Hermitian square root of R^{-1}. Then the following equivalences can be readily verified:

$$\begin{aligned} & R - \sigma_s^2 a(\theta) a^*(\theta) \geq 0 \\ & \quad \Longleftrightarrow \; I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2} \geq 0 \\ & \quad \Longleftrightarrow \; 1 - \sigma_s^2 a^*(\theta) R^{-1} a(\theta) \geq 0 \\ & \quad \Longleftrightarrow \; \sigma_s^2 \leq \frac{1}{a^*(\theta) R^{-1} a(\theta)} \end{aligned} \tag{6.5.33}$$

The third line in equation (6.5.33) follows from the fact that the eigenvalues of the matrix $I - \sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2}$ are equal to 1 minus the eigenvalues of $\sigma_s^2 R^{-1/2} a(\theta) a^*(\theta) R^{-1/2}$ (see Result R5 in Appendix A), and the latter eigenvalues are given by $\sigma_s^2 a^*(\theta) R^{-1} a(\theta)$ (which is the trace of the previous matrix) along with $(m-1)$ zeroes. From (6.5.33), we can see that the Capon spectral estimate is the solution to the problem (6.5.32) as well.

The equivalence between the formulation of the Capon method in (6.5.32) and the standard formulation in Section 6.3.2 can also be shown as follows: The constraint in (6.5.32) is equivalent to the requirement that

$$h^*[R - \sigma_s^2 a(\theta) a^*(\theta)]\, h \geq 0 \text{ for any } h \in \mathbf{C}^{m \times 1} \tag{6.5.34}$$

which, in turn, is equivalent to

$$\begin{aligned} &h^*[R - \sigma_s^2 a(\theta) a^*(\theta)]\, h \geq 0 \\ &\qquad \text{for any } h \text{ such that } h^* a(\theta) = 1 \end{aligned} \tag{6.5.35}$$

Clearly, (6.5.34) implies (6.5.35). To also show that (6.5.35) implies (6.5.34), let h be such that $h^* a(\theta) = \alpha \neq 0$; then h/α^* satisfies $(h/\alpha^*)^* a(\theta) = 1$ and, hence, by the assumption that (6.5.35) holds,

$$\frac{1}{|\alpha|^2} h^*[R - \sigma_s^2 a(\theta) a^*(\theta)] h \geq 0$$

which shows that (6.5.35) implies (6.5.34) for any h satisfying $h^* a(\theta) \neq 0$. Now, if h is such that $h^* a(\theta) = 0$, then

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

because $R > 0$ by assumption. This observation concludes the proof that (6.5.34) is equivalent to (6.5.35).

Using the equivalence of (6.5.34) and (6.5.35), we can rewrite (6.5.34) as follows:

$$h^* R h \geq \sigma_s^2 \quad \text{for any } h \text{ such that } h^* a(\theta) = 1 \tag{6.5.36}$$

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

$$\sigma_s^2 = \min_h h^* R h \quad \text{subject to } h^* a(\theta) = 1$$

which coincides with the standard formulation of the Capon method in (6.3.24).

The formulation of the Capon method in (6.5.32) will be used in Complement 6.5.4 to extend the method to the case where the direction vector $a(\theta)$ is known only imprecisely.

6.5.4 Capon Method for Uncertain Direction Vectors

The Capon method has better resolution and much better interference rejection capability (i.e., much lower leakage) than the beamforming method, *provided that the direction vector, $a(\theta)$, is*

known accurately. However, whenever the knowledge of $a(\theta)$ is imprecise, the performance of the Capon method could become worse than that of the beamforming method. To see why this is so, consider a scenario in which the problem is to estimate the power coming from a source with DOA assumed to be equal to θ_0. Let us assume that, in actuality, the true DOA of the source is $\theta_0 + \Delta$. For the Capon beamformer pointed toward θ_0, the source of interest (located at $\theta_0 + \Delta$) will play the role of an interference and will be attenuated. Consequently, the power of the signal of interest will be underestimated; the larger Δ is, the larger the underestimation error. Because steering-vector errors are common in applications, it follows that a *robust version of the Capon method* (i.e., one that is as insensitive to steering-vector errors as possible) would be highly desirable.

In this complement, we will present *an extension of the Capon method to the case of uncertain direction vectors*. Specifically, we will assume that the only knowledge we have about $a(\theta)$ is that it belongs to the uncertainty ellipsoid

$$(a - \bar{a})^* C^{-1} (a - \bar{a}) \leq 1 \tag{6.5.37}$$

where the vector $\bar{a}$ and the positive definite matrix C are given. Note that both a and $\bar{a}$, as well as C, usually depend on θ; however, for the sake of notational convenience, we drop the θ dependence of these variables.

In some applications, there will be too little available information about the errors in the steering vector to make a competent choice of the full matrix C in (6.5.37). In such cases, we may simply set $C = \varepsilon I$, so that (6.5.37) becomes

$$\|a - \bar{a}\|^2 \leq \varepsilon \tag{6.5.38}$$

where ε is a positive number. Let a_0 denote the true (and unknown) direction vector, and let $\varepsilon_0 = \|a_0 - \bar{a}\|^2$ where, as before, $\bar{a}$ is the assumed direction vector. Ideally, we should choose $\varepsilon = \varepsilon_0$. However, it can be shown (see [Stoica, Wang, and Li 2003], [Li, Stoica, and Wang 2003]) that the performance of the robust Capon method remains almost unchanged when ε is varied in a relatively large interval around ε_0.

As already stated, our goal here is to obtain a robust Capon method that is insensitive to errors in the direction (or steering) vector. We will do so by combining the covariance-fitting formulation in (6.5.32) for the standard Capon method with the steering uncertainty set in (6.5.37). Hence, we aim to derive estimates of *both* σ_s^2 *and* a by solving the following constrained covariance-fitting problem:

$$\max_{a,\sigma_s^2} \sigma_s^2 \quad \text{subject to: } \begin{array}{l} R - \sigma_s^2 a a^* \geq 0 \\ (a - \bar{a})^* C^{-1} (a - \bar{a}) \leq 1 \end{array} \tag{6.5.39}$$

To avoid the trivial solution ($a \to 0, \sigma_s^2 \to \infty$), we assume that $a = 0$ does not belong to the uncertainty ellipsoid in (6.5.39), or, equivalently, that

$$\bar{a}^* C^{-1} \bar{a} > 1 \tag{6.5.40}$$

(which is a regularity condition).

Because both σ_s^2 and a are considered to be free parameters in the previous fitting problem, there is a scaling ambiguity in the signal covariance term in (6.5.39), in the sense that both (σ_s^2, a) and $(\sigma_s^2/\mu, \mu^{1/2}a)$ for any $\mu > 0$ give the same covariance term $\sigma_s^2 aa^*$. To eliminate this ambiguity, we can use the knowledge that the true steering vector satisfies the following condition (see (6.3.11)):

$$a^*a = m \tag{6.5.41}$$

However, the constraint in (6.5.41) is nonconvex and so makes the combined problem (6.5.39) and (6.5.41) somewhat more difficult to solve than (6.5.39). On the other hand, (6.5.39) (without (6.5.41)) can be solved quite efficiently, as we show next. To take advantage of this fact, we can make use of (6.5.41) to eliminate the scaling ambiguity in the following pragmatic way:

- Obtain the solution $(\tilde{\sigma}_s^2, \tilde{a})$ of (6.5.39).
- Obtain an estimate of a that satisfies (6.5.41) by scaling $\tilde{a}$,

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

and a corresponding estimate of σ_s^2 by scaling $\tilde{\sigma}_s^2$, such that the signal covariance term is left unchanged (i.e., $\tilde{\sigma}_s^2\tilde{a}\tilde{a}^* = \hat{\sigma}_s^2\hat{a}\hat{a}^*$), which gives

$$\hat{\sigma}_s^2 = \tilde{\sigma}_s^2\frac{\|\tilde{a}\|^2}{m} \tag{6.5.42}$$

To derive the solution $(\tilde{\sigma}_s^2, \tilde{a})$ of (6.5.39), we first note that, for any fixed a, the maximizing σ_s^2 is given by

$$\tilde{\sigma}_s^2 = \frac{1}{a^*R^{-1}a} \tag{6.5.43}$$

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

$$\min_a a^*R^{-1}a \quad \text{subject to: } (a-\bar{a})^*C^{-1}(a-\bar{a}) \le 1 \tag{6.5.44}$$

Under the regularity condition in (6.5.40), the solution $\tilde{a}$ to (6.5.44) will occur on the boundary of the constraint set; therefore, we can reformulate (6.5.44) as the following quadratic problem with a quadratic equality constraint:

$$\min_a a^*R^{-1}a \quad \text{subject to: } (a-\bar{a})^*C^{-1}(a-\bar{a}) = 1 \tag{6.5.45}$$

This problem can be solved efficiently by using the Lagrange multiplier approach—see [Li, Stoica, and Wang 2003]. In the remaining part of this complement, we derive the Lagrange-multiplier solver in [Li, Stoica, and Wang 2003], but in a more self-contained way.

To simplify the notation, consider (6.5.45) with $C = \varepsilon I$ as in (6.5.38):

$$\min_a a^* R^{-1} a \quad \text{subject to: } \|a - \bar{a}\|^2 = \varepsilon \tag{6.5.46}$$

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

$$x = a - \bar{a} \tag{6.5.47}$$

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

$$\min_x \left[x^* R^{-1} x + x^* R^{-1} \bar{a} + \bar{a}^* R^{-1} x \right] \quad \text{subject to: } \|x\|^2 = \varepsilon \tag{6.5.48}$$

The constraint in (6.5.48) makes the x that solves (6.5.48) also a solution to the problem

$$\min_x \left[x^* (R^{-1} + \lambda I) x + x^* R^{-1} \bar{a} + \bar{a}^* R^{-1} x \right] \quad \text{subject to: } \|x\|^2 = \varepsilon \tag{6.5.49}$$

where λ is an arbitrary constant. Let us consider a particular choice of λ that is a solution of the equation

$$\bar{a}^* (I + \lambda R)^{-2} \bar{a} = \varepsilon \tag{6.5.50}$$

and is also such that

$$R^{-1} + \lambda I > 0 \tag{6.5.51}$$

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

$$x = -\left(R^{-1} + \lambda I\right)^{-1} R^{-1} \bar{a} = -(I + \lambda R)^{-1} \bar{a} \tag{6.5.52}$$

and it satisfies the constraint in (6.5.49) (*cf.* (6.5.50)). It follows that x in (6.5.52) with λ given by (6.5.50) and (6.5.51) is the solution to (6.5.49) (and, hence, to (6.5.48)). Hence, what is left to explain is how to solve (6.5.50) under the condition (6.5.51) in an efficient manner; we will do that next.

Let

$$R = U \Lambda U^* \tag{6.5.53}$$

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

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}; \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \tag{6.5.54}$$

Also, let

$$b = U^* \bar{a} \tag{6.5.55}$$

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

$$g(\lambda) \triangleq \bar{a}^* [I + \lambda R]^{-2} \bar{a} = \bar{a}^* \left[U(I + \lambda\Lambda)U^*\right]^{-2} \bar{a}$$
$$= b^*(I + \lambda\Lambda)^{-2} b = \sum_{k=1}^{m} \frac{|b_k|^2}{(1 + \lambda\lambda_k)^2} \tag{6.5.56}$$

where b_k is the kth element of the vector b. Note that

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

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

$$(1 + \lambda\lambda_k)^2 > 1 \tag{6.5.58}$$

for some value of k. At the same time, λ should (see (6.5.51)) be such that

$$R^{-1} + \lambda I > 0 \iff I + \lambda R > 0$$
$$\iff 1 + \lambda\lambda_k > 0 \text{ for } k = 1, \ldots, m \tag{6.5.59}$$

It follows from (6.5.58) and (6.5.59) that $1 + \lambda\lambda_k > 1$ for at least one value of k, which implies that

$$\lambda > 0 \tag{6.5.60}$$

This inequality sets a lower bound on the solution to (6.5.50). To refine this lower bound, and also to obtain an upper bound, first observe that $g(\lambda)$ is a *monotonically decreasing function of* λ for $\lambda > 0$. Furthermore, for

$$\lambda_L = \frac{\|\bar{a}\| - \sqrt{\varepsilon}}{\lambda_1\sqrt{\varepsilon}} \tag{6.5.61}$$

we have that

$$g(\lambda_L) \geq \frac{1}{(1 + \lambda_L\lambda_1)^2}\|b\|^2 = \frac{\varepsilon}{\|\bar{a}\|^2}\|\bar{a}\|^2 = \varepsilon \tag{6.5.62}$$

Similarly, for

$$\lambda_U = \frac{\|\bar{a}\| - \sqrt{\varepsilon}}{\lambda_m\sqrt{\varepsilon}} \geq \lambda_L \tag{6.5.63}$$

we can verify that

$$g(\lambda_U) \leq \frac{1}{(1 + \lambda_U\lambda_m)^2}\|b\|^2 = \varepsilon \tag{6.5.64}$$

Summarizing the previous facts, it follows that *equation* (6.5.50) *has a unique solution for* λ *that satisfies (6.5.51), which belongs to the interval* $[\lambda_L, \lambda_U] \subset (0, \infty)$. With this observation, the derivation of the robust version of the Capon method is complete. The following is a *step-by-step summary of the Robust Capon algorithm*.

The Robust Capon Algorithm

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

Step 2. Solve the equation $g(\lambda) = \varepsilon$ for λ (using, e.g., a Newton method, along with the fact that there is a unique solution in the interval $[\lambda_L, \lambda_U]$).

Step 3. Compute (*cf.* (6.5.47), (6.5.52), (6.5.53))

$$\tilde{a} = \bar{a} - U(I + \lambda\Lambda)^{-1} b \tag{6.5.65}$$

and, finally, compute the power estimate (see (6.5.42) and (6.5.43))

$$\hat{\sigma}_s^2 = \frac{\tilde{a}^* \tilde{a}}{m\, \tilde{a}^* U \Lambda^{-1} U^* \tilde{a}} \tag{6.5.66}$$

where, from (6.5.65), $U^* \tilde{a} = b - (I + \lambda\Lambda)^{-1} b$.

The bulk of the computation in the algorithm involves computing the EVD of R, which requires $\mathcal{O}(m^3)$ arithmetic operations. Hence, the computational complexity of the Robust Capon method is comparable to that of the standard Capon method. We refer the reader to [Li, Stoica, and Wang 2003] and also to [Stoica, Wang, and Li 2003] for further computational considerations and insights, and for many numerical examples illustrating the good performance of the Robust Capon method, including its insensitivity to the choice of ε in (6.5.38) or C in (6.5.37).

6.5.5 Capon Method with Noise-Gain Constraint

As was explained in Complement 6.5.4, the Capon method performs poorly as a power estimator in the presence of steering-vector errors. (Yet, it could perform fairly well as a DOA estimator, provided that the SNR is reasonably large; see [Cox 1973; Li, Stoica, and Wang 2003] and references therein.) The same happens when the number of snapshots, N, is relatively small, such as when N is equal to or only slightly larger than the number of sensors, m. In fact, there is a close relationship between the cases of steering-vector errors and those of small-sample errors—see, for example, [Feldman and Griffiths 1994]. More precisely, the sampling estimation errors of the covariance matrix can be viewed as steering-vector errors in a corresponding theoretical covariance matrix, and vice versa. For example, consider a uniform linear array, and assume that the source signals are uncorrelated with one another. In this case, the theoretical covariance matrix R of the array output is Toeplitz. Assume that the sample covariance matrix $\hat{R}$ is also Toeplitz. According to the Carathéodory parameterization of Toeplitz matrices (see Complement 4.9.2), we

can view $\hat{R}$ as being the *theoretical* covariance matrix associated with a fictitious ULA on which uncorrelated signals impinge, but the powers and DOAs of the latter signals are different from those of the actual signals. Hence, the small sample estimation errors in $\hat{R}$ can be viewed as being due to steering-vector errors in a corresponding theoretical covariance matrix.

The robust Capon method (RCM) presented in Complement 6.5.4 significantly outperforms the standard Capon method (CM) in power-estimation applications in which the sample length is insufficient for accurate estimation of R or in which the steering vector is known imprecisely. The RCM was introduced in [Stoica, Wang, and Li 2003; Li, Stoica, and Wang 2003]. An earlier approach, whose goal is also to enhance the performance of CM in the presence of sampling-estimation errors or steering-vector mismatch, is the so-called *diagonal-loading* approach (see, e.g., [Hudson 1981; Van Trees 2002] and references therein). The main idea of diagonal loading is to replace R in the Capon formula for the spatial filter h, (6.3.24), by the matrix

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

where the diagonal-loading factor $\lambda > 0$ is a user-selected parameter. The filter vector h so obtained is given by

$$h = \frac{(R + \lambda I)^{-1} a}{a^*(R + \lambda I)^{-1} a} \tag{6.5.68}$$

The use of the diagonally loaded matrix in (6.5.67) instead of R is the reason for the name of the approach based on (6.5.68). The symbol R in this complement refers either to a theoretical covariance matrix or to a sample covariance matrix.

There have been several rules proposed in the literature for choosing the parameter λ in (6.5.68). Most of these rules choose λ in a rather *ad hoc* and data-independent manner. As illustrated in [Li, Stoica, and Wang 2003] and its references, a data-independent selection of the diagonal-loading factor *cannot* improve the performance for a reasonably large range of SNR values. Hence, a *data-dependent choice of* λ is desired.

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

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

where the constant c must be chosen by the user. Let us explain briefly why choosing λ via (6.5.69) makes sense intuitively. Assume that the array output vector contains a spatially white noise component whose covariance matrix is proportional to I (see (6.4.1)). Then the power at the output of the spatial filter h due to the noise component is $\|h\|^2$; for this reason, $\|h\|^2$ is sometimes called *the (white) noise gain* of h. In scenarios with a large number of (possibly closely spaced) source signals, the Capon spatial filter h in (6.3.24) could run out of "degrees of freedom" and hence not pay enough attention to the noise in the data (unless the SNR is very low). The result is a relatively high noise gain, $\|h\|^2$, which may well degrade the accuracy of signal power estimation. To prevent this from happening, it makes sense to limit $\|h\|^2$ as in (6.5.69). By doing so, we are left with the problem of choosing c. The choice of c might be easier than the direct

choice of λ in (6.5.68), yet it is far from trivial, and, in fact, clear-cut rules for selecting c are hardly available. In particular, a "too small" value of c could limit the noise gain unnecessarily and result in decreased resolution and increased leakage.

In this complement, we will show that the spatial filter of the diagonally loaded Capon method in (6.5.68), (6.5.69) is the solution to the following design problem:

$$\min_h h^*Rh \quad \text{subject to: } h^*a = 1 \text{ and } \|h\|^2 \le c \tag{6.5.70}$$

Because (6.5.70) is obtained by adding the noise-gain constraint $\|h\|^2 \le c$ to the standard Capon problem in (6.3.23), we will call the method that follows from (6.5.70) the *constrained Capon method* (CCM). The fact that (6.5.68), (6.5.69) is the solution to (6.5.70) is well known from the previous literature (see, e.g., [Hudson 1981]). However, we present a rigorous and more thorough analysis of this solution. As a by-product, the analysis that follows also suggests some guidelines for choosing the user parameter c in (6.5.69). Note that, in general, a, c, and h in (6.5.70) *depend on the DOA* θ; to simplify notation, we will omit the functional dependence on θ here.

It is interesting to observe that *the RCM, described in Complement 6.5.4, can also be cast into a diagonal-loading framework*. To see this, first note from (6.5.47) and (6.5.52) that the steering-vector estimate used in the RCM is given by

$$\begin{aligned} a &= \bar{a} - (I + \lambda R)^{-1}\bar{a} = (I + \lambda R)^{-1}\left[(I + \lambda R) - I\right]\bar{a} \\ &= \left(\frac{1}{\lambda}R^{-1} + I\right)^{-1}\bar{a} \end{aligned} \tag{6.5.71}$$

The RCM estimates the signal power by

$$\frac{1}{a^*R^{-1}a} \tag{6.5.72}$$

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

$$h = \frac{R^{-1}a}{a^*R^{-1}a} \tag{6.5.73}$$

hence, (6.5.72) can be viewed as being obtained by the (implicit) use of the spatial filter in (6.5.71), (6.5.73). Inserting (6.5.71) into (6.5.73), we obtain

$$h = \frac{\left(R + \frac{1}{\lambda}I\right)^{-1}a}{a^*\left[\left(R + \frac{1}{\lambda}I\right)R^{-1}\left(R + \frac{1}{\lambda}I\right)\right]^{-1}a} \tag{6.5.74}$$

which, except for the scalar in the denominator, has the form in (6.5.68) of the spatial filter used by the diagonal-loading approach. Note that the diagonal-loading factor, $1/\lambda$, in (6.5.74) is data dependent. Furthermore, the selection of λ in the RCM (see Complement 6.5.4 for details on this

aspect) relies entirely on information about the uncertainty set of the steering vector, as defined, for instance, by the sphere with radius $\varepsilon^{1/2}$ in (6.5.38). Such information is more readily available in applications than is information that would help the user select the noise-gain constraint c in the CCM. Indeed, in many applications, we should be able to make *a more competent guess about ε than about c* (for all DOAs of interest in the analysis). This appears to be a significant advantage of RCM over CCM, despite the fact that both methods can be interpreted as data-dependent diagonal-loading approaches.

Remark: The reader might have noted by now that the CCM problem in (6.5.70) is similar to the combined RCM problem in (6.5.44), (6.5.41) discussed in Complement 6.5.4. This observation has two consequences. First, it follows that the combined RCM design problem in (6.5.44), (6.5.41) could be solved by an algorithm similar to the one presented below for solving the CCM problem; indeed, this is the case, as is shown in [Li, Stoica, and Wang 2004]. Second, the CCM problem (6.5.70) and the combined RCM problem (6.5.44), (6.5.41) both have two constraints and are more complicated than the RCM problem (6.5.44), which has only one constraint. Hence, the CCM algorithm described below will be (slightly) more involved computationally than the RCM algorithm outlined in Complement 6.5.4. ■

We begin the analysis of the CCM problem in (6.5.70) by deriving a feasible range for the user parameter c. Let S denote the set of vectors h that satisfy both constraints in (6.5.70):

$$S = \{h \mid h^*a = 1 \text{ and } \|h\|^2 \le c\} \tag{6.5.75}$$

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

$$1 = \left|h^*a\right|^2 \le \|h\|^2\|a\|^2 \le cm \implies c \ge \frac{1}{m} \tag{6.5.76}$$

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

$$\|a\|^2 = m \tag{6.5.77}$$

The inequality in (6.5.76) sets a lower bound on c; otherwise, S is empty. To obtain an upper bound, we can argue as follows: The vector h used in the CM has the norm

$$\|h_{\mathrm{CM}}\|^2 = \frac{a^*R^{-2}a}{\left(a^*R^{-1}a\right)^2} \tag{6.5.78}$$

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

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

Note that, if c does not satisfy (6.5.79), then the CM spatial filter h satisfies both constraints in (6.5.70) and, hence, *is the solution to the CCM problem*. Combining (6.5.76) and (6.5.79) yields

the following interval for c:

$$\boxed{c \in \left[\frac{1}{m}, \frac{a^*R^{-2}a}{\left(a^*R^{-1}a\right)^2}\right]} \tag{6.5.80}$$

As with (6.5.53), let

$$R = U\Lambda U^* \tag{6.5.81}$$

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

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}; \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \tag{6.5.82}$$

Now,

$$\frac{a^*R^{-2}a}{\left[a^*R^{-1}a\right]^2} \leq \frac{\|a\|^2/\lambda_m^2}{\left[\|a\|^2/\lambda_1\right]^2} = \frac{\lambda_1^2}{m\lambda_m^2} \tag{6.5.83}$$

so it follows from (6.5.79) that c also satisfies

$$mc < \frac{\lambda_1^2}{\lambda_m^2} \tag{6.5.84}$$

The above inequality will be useful later on.

Next, let us define the function

$$g(h, \lambda, \mu) = h^*Rh + \lambda(\|h\|^2 - c) + \mu(-h^*a - a^*h + 2) \tag{6.5.85}$$

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

$$\lambda > 0 \tag{6.5.86}$$

Remark: We note, in passing, that λ and μ are the so-called Lagrange multipliers and that $g(h, \lambda, \mu)$ is the so-called Lagrangian function associated with the CCM problem in (6.5.70); however, to make the following derivation as self-contained as possible, we will not explicitly use any result from Lagrange-multiplier theory. ■

Evidently, by the definition of $g(h, \lambda, \mu)$, we have that

$$g(h, \lambda, \mu) \leq h^*Rh \quad \text{for any } h \in S \tag{6.5.87}$$

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

$$\begin{aligned} & h^*(R+\lambda I)h - \mu h^* a - \mu a^* h \\ & \quad = \left[h - \mu(R+\lambda I)^{-1}a\right]^* (R+\lambda I)\left[h - \mu(R+\lambda I)^{-1}a\right] \\ & \qquad - \mu^2 a^*(R+\lambda I)^{-1}a \end{aligned} \tag{6.5.88}$$

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

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

Let us choose μ such that (6.5.89) satisfies the first constraint in (6.5.70):

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

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

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

We will show shortly that this equation has a unique solution $\hat{\lambda} > 0$ for *any* c satisfying (6.5.80). Before doing so, we remark on the following important fact: Inserting (6.5.90) into (6.5.89), we get the diagonally loaded version of the Capon method (see (6.5.68))—that is,

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

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

$$g\left(\hat{h}(\hat{\lambda}, \hat{\mu}), \hat{\lambda}, \hat{\mu}\right) = \hat{h}^*(\hat{\lambda}, \hat{\mu})R\hat{h}(\hat{\lambda}, \hat{\mu}) \le h^* R h \quad \text{for any } h \in S \tag{6.5.93}$$

From (6.5.93), we conclude that (6.5.92) *is the (unique) solution to the CCM problem.*

It remains to show that, indeed, equation (6.5.91) has a unique solution $\hat{\lambda} > 0$ under (6.5.80) and also to provide a computationally convenient way of finding $\hat{\lambda}$. Towards that end, we use the EVD of R in (6.5.91) (with the hat on $\hat{\lambda}$ omitted, for notational simplicity) to rewrite (6.5.91) as follows:

$$f(\lambda) = c \tag{6.5.94}$$

where

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

and where b_k is the kth element of the vector

$$b = U^*a \tag{6.5.96}$$

Differentiation of (6.5.95) with respect to λ yields

$$\begin{aligned} f'(\lambda) = & \left\{ -2\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^2 \right. \\ & \left. +2\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right] \right\} \\ & \cdot \frac{1}{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^4} \\ = & -2\left\{ \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right] - \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]^2 \right\} \\ & \cdot \frac{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]}{\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right]^4} \end{aligned} \tag{6.5.97}$$

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

$$\begin{aligned} \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^2}\right]^2 &= \left[\sum_{k=1}^{m} \frac{|b_k|}{(\lambda_k+\lambda)^{3/2}} \frac{|b_k|}{(\lambda_k+\lambda)^{1/2}}\right]^2 \\ &< \left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)^3}\right]\left[\sum_{k=1}^{m} \frac{|b_k|^2}{(\lambda_k+\lambda)}\right] \end{aligned} \tag{6.5.98}$$

Hence,

$$f'(\lambda) < 0 \text{ for any } \lambda > 0 \quad (\text{and } \lambda_k \neq \lambda_p \text{ for at least one pair } k \neq p) \tag{6.5.99}$$

which means that $f(\lambda)$ is a *monotonically strictly decreasing function* for $\lambda > 0$. Combining this observation with the fact that $f(0) > c$ (see (6.5.79)) shows that, indeed, the equation $f(\lambda) = c$ in (6.5.91) has a *unique solution* for $\lambda > 0$.

For efficiently solving the equation $f(\lambda) = c$, an upper bound on λ would also be useful. Such a bound can be obtained from (6.5.95) as follows: A simple calculation shows that

$$c = f(\lambda) < \frac{\dfrac{\|b\|^2}{(\lambda_m + \lambda)^2}}{\dfrac{\|b\|^4}{(\lambda_1 + \lambda)^2}} = \frac{(\lambda_1 + \lambda)^2}{m(\lambda_m + \lambda)^2}$$

$$\Longrightarrow mc(\lambda_m + \lambda)^2 < (\lambda_1 + \lambda)^2 \tag{6.5.100}$$

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

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

Note that both the numerator and the denominator in (6.5.101) are positive; see (6.5.76) and (6.5.84).

The derivation of the constrained Capon method is now complete. The following is *a step-by-step summary of the CCM*.

The Constrained Capon Algorithm

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

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

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

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

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

To conclude this complement, we note that the CCM algorithm is quite similar to the RCM algorithm presented in Complement 6.5.4. The only differences are that the equation for λ

associated with the CCM is slightly more complicated and, more importantly, that it is harder to select the c needed in the CCM (for any DOA of interest) than it is to select ε in the RCM. As we have shown, for CCM one should choose c in the interval (6.5.80). Note that for $c = 1/m$ we get $\lambda \to \infty$ and $h = a/m$, which is the beamforming method. For $c = a^*R^{-2}a/(a^*R^{-1}a)^2$, we obtain $\lambda = 0$ and $h = h_{\mathrm{CM}}$, which is the standard Capon method. Values of c between these two extremes should be chosen in an application-dependent manner.

6.5.6 Spatial Amplitude and Phase Estimation (APES)

As was explained in Section 6.3.2, the Capon method estimates the spatial spectrum by using a spatial filter that passes the signal impinging on the array from direction θ in a distortionless manner and at the same time attenuates signals with DOAs different from θ as much as possible. The Capon method for temporal spectral analysis is based on exactly the same idea (see Section 5.4), as is the temporal APES method described in Complement 5.6.4. In this complement, we will present an extension of APES that can be used for spatial spectral analysis.

Let θ denote a generic DOA, and consider the equation (6.2.19),

$$y(t) = a(\theta)s(t) + e(t), \qquad t = 1, \ldots, N \tag{6.5.102}$$

that describes the array output, $y(t)$, as a function of a signal, $s(t)$, possibly impinging on the array from a DOA equal to θ, and a term, $e(t)$, that includes noise along with any other signals whose DOAs are different from θ. We assume that the array is *uniform and linear*, in which case $a(\theta)$ is given by

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

where m denotes the number of sensors in the array, and $\omega_s = (\omega_c d \sin\theta)/c$ is the spatial frequency (see (6.2.26) and (6.2.27)). As we will explain later, the spatial extension of APES presented in this complement appears to perform well only in the case of ULAs. This is a limitation, but it is not a serious one, because there are techniques that can be used to approximately transform the direction vector of a general array into the direction vector of a fictitious ULA (see, for example, [Doron, Doron, and Weiss 1993]). Such a technique performs a relatively simple DOA-independent linear transformation of the array output snapshots; these linearly transformed snapshots can then be used as the input to the spatial APES method presented here. (See [Abrahamsson, Jakobsson, and Stoica 2004] for details on how to use the spatial APES approach of this complement for arrays that are not uniform and linear.)

Let σ_s^2 denote the power of the signal $s(t)$ in (6.5.102), which is the main parameter we want to estimate; note that the estimated signal power $\hat{\sigma}_s^2$, as a function of θ, provides an estimate of the spatial spectrum. In this complement, we assume that $\{s(t)\}_{t=1}^N$ is an unknown *deterministic* sequence; hence, we define σ_s^2 as

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

An important difference between equation (6.5.102) and its temporal counterpart (see, e.g., equation (5.6.80) in Complement 5.6.6) is that, in (6.5.102), the signal $s(t)$ is *completely unknown*, whereas, in the temporal case, we have $s(t) = \beta e^{i\omega t}$ and only the amplitude is unknown. Because of this difference, the use of the APES principle for spatial spectral estimation is somewhat different from its use for temporal spectral estimation.

Remark: We remind the reader that $\{s(t)\}_{t=1}^{N}$ is assumed to be an unknown deterministic sequence here. The case in which $\{s(t)\}$ is assumed to be stochastic is considered in Complement 6.5.3. Interestingly, application of the APES principle in the stochastic signal case leads to the (standard) Capon method! ■

Let $\bar{m} < m$ be an integer, and define the following two vectors:

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

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

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

$$L = m - \bar{m} + 1 \tag{6.5.107}$$

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

The assumed ULA structure means that the direction subvectors $\{\bar{a}_k\}$ satisfy the following relations:

$$\bar{a}_k = e^{-i(k-1)\omega_s}\bar{a}_1, \qquad k = 2, \ldots, L \tag{6.5.108}$$

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

$$\bar{y}_k(t) = \bar{a}_k s(t) + \bar{e}_k(t) = e^{-i(k-1)\omega_s}\bar{a}_1 s(t) + \bar{e}_k(t) \tag{6.5.109}$$

where $\bar{e}_k(t)$ is a noise vector defined similarly to $\bar{y}_k(t)$. Let h denote the $(\bar{m} \times 1)$ coefficient vector of a spatial filter that is applied to $\{e^{i(k-1)\omega_s}\bar{y}_k(t)\}_{k=1}^{L}$. Then it follows from (6.5.109) that h passes the signal $s(t)$ in each of these data sets in a distortionless manner if and only if

$$h^*\bar{a}_1 = 1 \tag{6.5.110}$$

Using the preceding observations along with the APES principle presented in Complement 5.6.4, we can determine both the spatial filter h and an estimate of the complex-valued sequence $\{s(t)\}_{t=1}^{N}$ (we estimate both amplitude and phase—recall that APES stands for Amplitude and Phase ESti-mation) by solving the following linearly constrained least-squares (LS) problem:

$$\min_{h;\{s(t)\}} \sum_{t=1}^{N}\sum_{k=1}^{L}\left|h^*\bar{y}_k(t)e^{i(k-1)\omega_s} - s(t)\right|^2 \quad \text{subject to: } h^*\bar{a}_1 = 1 \tag{6.5.111}$$

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

To derive a solution to (6.5.111), let

$$g(t) = \frac{1}{L}\sum_{k=1}^{L} \bar{y}_k(t)e^{i(k-1)\omega_s} \tag{6.5.112}$$

and observe that

$$\begin{aligned}
&\frac{1}{L}\sum_{k=1}^{L}\left|h^*\bar{y}_k(t)e^{i(k-1)\omega_s} - s(t)\right|^2 \\
&\quad = |s(t)|^2 + h^*\left[\frac{1}{L}\sum_{k=1}^{L}\bar{y}_k(t)\bar{y}_k^*(t)\right]h - h^*g(t)s^*(t) - g^*(t)hs(t) \\
&\quad = h^*\left[\frac{1}{L}\sum_{k=1}^{L}\bar{y}_k(t)\bar{y}_k^*(t)\right]h - h^*g(t)g^*(t)h + \left|s(t) - h^*g(t)\right|^2
\end{aligned} \tag{6.5.113}$$

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

$$\boxed{\hat{s}(t) = h^*g(t)} \tag{6.5.114}$$

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

$$\min_h h^*\hat{Q}h \qquad \text{subject to: } h^*\bar{a}_1 = 1 \tag{6.5.115}$$

where

$$\begin{aligned}
\hat{Q} &= \hat{R} - \hat{G} \\
\hat{R} &= \frac{1}{N}\sum_{t=1}^{N}\frac{1}{L}\sum_{k=1}^{L}\bar{y}_k(t)\bar{y}_k^*(t) \\
\hat{G} &= \frac{1}{N}\sum_{t=1}^{N}g(t)g^*(t)
\end{aligned} \tag{6.5.116}$$

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

$$\hat{h} = \frac{\hat{Q}^{-1}\bar{a}_1}{\bar{a}_1^*\hat{Q}^{-1}\bar{a}_1} \tag{6.5.117}$$

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

$$\hat{\sigma}_s^2 = \frac{1}{N}\sum_{t=1}^{N}\left|\hat{s}(t)\right|^2 = \hat{h}^*\hat{G}\hat{h} \tag{6.5.118}$$

This equation, as a function of DOA θ, provides an estimate of the spatial spectrum.

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

$$\hat{Q} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{L}\sum_{k=1}^{L}\left[e^{i(k-1)\omega_s}\bar{y}_k(t) - g(t)\right]\left[e^{i(k-1)\omega_s}\bar{y}_k(t) - g(t)\right]^* \tag{6.5.119}$$

It follows from (6.5.119) that $\hat{Q}$ is always positive semidefinite. For $L = 1$ (or, equivalently, $\bar{m} = m$), we have $\hat{Q} = 0$ because $g(t) = \bar{y}_1(t)$ for $t = 1, \ldots, N$. Thus, for $L = 1$, (6.5.117) is not valid. This is expected: indeed, for $L = 1$ we can make (6.5.111) equal to zero, for *any* h, by choosing $\hat{s}(t) = h^*\bar{y}_1(t)$; consequently, the problem of minimizing (6.5.111) with respect to $(h; \{s(t)\}_{t=1}^N)$ is underdetermined for $L = 1$, and, hence, an infinite number of solutions exist. To prevent this from happening, we should choose $L \geq 2$ (or, equivalently, $\bar{m} \leq m - 1$). For $L \geq 2$, the $(\bar{m} \times \bar{m})$ matrix $\hat{Q}$ is a sum of NL outer products; if $NL \geq \bar{m}$, which is a weak condition, $\hat{Q}$ is almost surely strictly positive definite and hence nonsingular.

From a performance point of view, it turns out that a good choice of $\bar{m}$ is its maximum possible value:

$$\bar{m} = m - 1 \qquad \Longleftrightarrow \qquad L = 2 \tag{6.5.120}$$

A numerical study of performance, reported in [Gini and Lombardini 2002], supports this choice of $\bar{m}$ and also suggests that the spatial APES method can outperform the Capon method both in spatial-spectrum estimation and in DOA-estimation applications. The APES spatial filter is, however, more difficult to compute than is the Capon spatial filter, as a result of the dependence of $\hat{Q}$ in (6.5.117) on the DOA.

In the remainder of this complement, we will explain why the APES method may be expected to outperform the Capon method. In doing so, we assume that $\bar{m} = m - 1$ (and thus $L = 2$), as in (6.5.120). Intuitively, this choice of $\bar{m}$ provides the APES filter with the maximum possible

number of degrees of freedom; hence, it makes sense that it should lead to better resolution and interference-rejection capability than would smaller values of $\bar{m}$. For $L = 2$, we have

$$g(t) = \frac{1}{2}[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)] \tag{6.5.121}$$

and, hence,

$$\begin{aligned}\hat{Q} &= \frac{1}{2N}\sum_{t=1}^{N}\left\{\frac{1}{4}[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)][\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)]^*\right. \\ &\qquad \left. + \frac{1}{4}[e^{i\omega_s}\bar{y}_2(t) - \bar{y}_1(t)][e^{i\omega_s}\bar{y}_2(t) - \bar{y}_1(t)]^*\right\} \\ &= \frac{1}{4N}\sum_{t=1}^{N}[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)][\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)]^*\end{aligned} \tag{6.5.122}$$

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

$$\min_h \sum_{t=1}^{N}\left|h^*\left[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)\right]\right|^2 \quad \text{subject to: } h^*\bar{a}_1 = 1 \tag{6.5.123}$$

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

$$\hat{s}(t) = \frac{1}{2}h^*\left[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)\right] \tag{6.5.124}$$

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

$$\min_h \sum_{t=1}^{N}\left|h^*y(t)\right|^2 \quad \text{subject to: } h^*a = 1 \tag{6.5.125}$$

and the Capon signal estimate is given by

$$\hat{s}(t) = h^*y(t) \tag{6.5.126}$$

To explain the main differences between the APES and Capon approaches, let us assume that, in addition to the signal of interest (SOI) $s(t)$ impinging on the array from the DOA under consideration θ, there is an interference signal $i(t)$ that impinges on the array from another DOA, denoted θ_i. We consider the situation in which only one interference signal is present, to simplify the discussion, but the case of multiple interference signals can be treated similarly. The array output vector in (6.5.102) and the subvectors in (6.5.109) become

$$y(t) = a(\theta)s(t) + b(\theta_i)i(t) + e(t) \tag{6.5.127}$$

$$\bar{y}_1(t) = \bar{a}_1(\theta)s(t) + \bar{b}_1(\theta_i)i(t) + \bar{e}_1(t) \tag{6.5.128}$$

$$\bar{y}_2(t) = \bar{a}_2(\theta)s(t) + \bar{b}_2(\theta_i)i(t) + \bar{e}_2(t) \tag{6.5.129}$$

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

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

(i) The SOI steering vector is known imprecisely—for example, as a result of pointing or calibration errors;
(ii) The SOI is highly correlated or coherent with the interference, which happens in multipath-propagation or smart-jamming scenarios.

To explain the difficulty of the Capon method in case (i), let us assume that the true steering vector of the SOI is $a_0 \neq a$. Then, by design, the Capon filter will be such that $|h^* a_0| \simeq 0$ (where $\simeq 0$ denotes a "small" value). Therefore, the SOI, whose steering vector is different from the assumed vector a, is treated as an interference signal and is attenuated or cancelled. As a consequence, the power of the SOI will be significantly underestimated, unless special measures are taken to make the Capon method robust against steering-vector errors (see Complements 6.5.4 and 6.5.5.)

The performance degradation of the Capon method in case (ii) is also easy to understand. Assume that the interference is coherent with the SOI and, hence, that $i(t) = \rho s(t)$ for some nonzero constant ρ. Then (6.5.127) can be rewritten as

$$y(t) = (a + \rho b)s(t) + e(t) \tag{6.5.130}$$

which shows that the SOI steering vector is given by $(a + \rho b)$ in lieu of the assumed vector a. Consequently, the Capon filter will by design be such that $|h^*(a + \rho b)| \simeq 0$, and therefore the SOI will be attenuated or cancelled in the filter output $h^* y(t)$, as in case (i). In fact, case (ii) can be considered as an extreme example of case (i), in which the SOI steering vector errors can be significant. Modifying the Capon method to work well in the case of coherent multipath signals is thus a more difficult problem than modifying it to be robust to small steering-vector errors.

Next, let us consider the APES method in case (ii). From (6.5.128) and (6.5.129), along with (6.5.108), we get

$$\begin{aligned}
&\left[\bar{y}_1(t) - e^{i\omega_s}\bar{y}_2(t)\right] \\
&\quad = \left(\bar{a}_1 - e^{i\omega_s}\bar{a}_2\right)s(t) + \left(\bar{b}_1 - e^{i\omega_s}\bar{b}_2\right)i(t) + \left[\bar{e}_1(t) - e^{i\omega_s}\bar{e}_2(t)\right] \\
&\quad = \left[1 - e^{i(\omega_s - \omega_i)}\right]\bar{b}_1 i(t) + \left[\bar{e}_1(t) - e^{i\omega_s}\bar{e}_2(t)\right]
\end{aligned} \tag{6.5.131}$$

and

$$\begin{aligned}
&\frac{1}{2}\left[\bar{y}_1(t) + e^{i\omega_s}\bar{y}_2(t)\right] \\
&\quad = \frac{1}{2}\left(\bar{a}_1 + e^{i\omega_s}\bar{a}_2\right)s(t) + \frac{1}{2}\left(\bar{b}_1 + e^{i\omega_s}\bar{b}_2\right)i(t) + \frac{1}{2}\left[\bar{e}_1(t) + e^{i\omega_s}\bar{e}_2(t)\right] \\
&\quad = \bar{a}_1 s(t) + \frac{1}{2}\left[1 + e^{i(\omega_s - \omega_i)}\right]\bar{b}_1 i(t) + \frac{1}{2}\left[\bar{e}_1(t) + e^{i\omega_s}\bar{e}_2(t)\right]
\end{aligned} \tag{6.5.132}$$

where $\omega_i = (\omega_c d \sin\theta_i)/c$ denotes the spatial frequency of the interference. It follows from (6.5.131) and the design criterion in (6.5.123) that the APES spatial filter will be such that

$$\left|1 - e^{i(\omega_s - \omega_i)}\right| \cdot \left|h^* \bar{b}_1\right| \simeq 0 \tag{6.5.133}$$

Hence, because the SOI is absent from the data vector in (6.5.131), the APES filter is able to cancel the interference only, despite the fact that the interference and the SOI are coherent. This interference-rejection property of the APES filter (i.e., $|h^* \bar{b}_1| \simeq 0$) is precisely what is needed when estimating the SOI from the data in (6.5.132).

To summarize, the APES method circumvents the problem in case (ii) by implicitly eliminating the signal from the data that is used to derive the spatial filter. However, if there is more than one coherent interference in the observed data, then APES also breaks down, in much the way that the Capon method does. The reason is that the vector multiplying $i(t)$ in (6.5.131) is no longer proportional to the vector multiplying $i(t)$ in (6.5.132); hence, a filter h that, by design, cancels the interference $i(t)$ in (6.5.131) is not guaranteed to have the desirable effect of cancelling $i(t)$ in (6.5.132); the details are left to the interested reader.

Remark: An argument similar to the one above explains why APES will not work well for non-ULA array geometries, in spite of the fact that it can be extended to such geometries in a relatively straightforward manner. Specifically, for non-ULA geometries, the steering vectors of the interference terms in the data sets used to obtain h and to estimate $s(t)$, respectively, are not proportional to one another. As a consequence, the design objective does not provide the APES filter with the desired capability of attenuating the interference terms in the data that is used to estimate $\{s(t)\}$. ■

Next consider the APES method in case (i). To simplify the discussion, let us assume that there are no calibration errors, but only a pointing error, so that the true spatial frequency of the SOI is $\omega_s^0 \neq \omega_s$. Then equation (6.5.131) becomes

$$\begin{aligned} \bar{y}_1(t) - e^{i\omega_s} \bar{y}_2(t) = &\left[1 - e^{i(\omega_s - \omega_s^0)}\right] \bar{a}_1^0 s(t) + \left[1 - e^{i(\omega_s - \omega_i)}\right] \bar{b}_1 i(t) \\ &+ \left[\bar{e}_1(t) - e^{i\omega_s} \bar{e}_2(t)\right] \end{aligned} \tag{6.5.134}$$

It follows that, in case (i), the APES spatial filter tends to cancel the SOI, in addition to cancelling the interference. However, the pointing errors are usually quite small; therefore, the residual term of $s(t)$ in (6.5.134) is small as well. Hence, the SOI might well pass through the APES filter (i.e., $|h^* \bar{a}_1^0|$ may be reasonably close to $|h^* \bar{a}_1| = 1$), because the filter uses most of its degrees of freedom to cancel the much stronger interference term in (6.5.134). As a consequence, APES is less sensitive to steering-vector errors than is the Capon method.

The previous discussion also explains why APES can provide *better power estimates* than the Capon method, even in "ideal" cases in which there are no multipath signals that are coherent with the SOI and no steering-vector errors, but the number of snapshots N is not very large. Indeed, as argued in Complement 6.5.5, the finite-sample effects associated with practical values of N can be viewed as inducing both correlation among the signals and steering-vector errors, to which the APES method is less sensitive than the Capon method, as has been explained.

We also note that the power of the elements of the noise vector in the data in (6.5.131) that is used to derive the APES filter is larger than the power of the noise elements in the raw data $y(t)$ that is used to compute the Capon filter. Somewhat counterintuitively, this is another potential advantage of the APES method over the Capon method. Indeed, the increased noise power in the data used by APES has a regularizing effect on the APES filter, which keeps the filter noise gain down, whereas the Capon filter is known to have a relatively large noise gain that can have a detrimental effect on signal-power estimation. (See Complement 6.5.5.)

On the downside, APES has been found to have a *slightly lower resolution* than the Capon method. (See, e.g., [Jakobsson and Stoica 2000].) Our previous discussion also provides a simple explanation of this result: when the interference and the SOI are closely spaced (i.e., when $\omega_s \simeq \omega_i$), the first factor in (6.5.133) becomes rather small and so might allow the second factor to increase somewhat. This explains how the beamwidth of the APES spatial filter could be larger than that of the Capon filter and, hence, why APES might have a slightly lower resolution.

6.5.7 The CLEAN Algorithm

The CLEAN algorithm is a *semiparametric method* that can be used for spatial spectral estimation. As we will see, this algorithm can be introduced in a nonparametric fashion (see [Högbom 1974]), yet its performance depends heavily on an implicit parametric assumption about the structure of the spatial covariance matrix; thus, CLEAN lies in between the class of nonparametric and parametric approaches, and it can be called a semiparametric approach.

There is a significant literature about CLEAN and its many applications in diverse areas, including array signal processing, image processing, and astronomy (see, for example, [Cornwell and Bridle 1996] and its references.) Our discussion of CLEAN will focus on its application to spatial spectral analysis and DOA estimation.

First, we present an *intuitive motivation* of CLEAN. Consider the beamforming spatial spectral estimate in (6.3.18), namely,

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

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

$$\hat{\theta}_1 = \arg\max_{\theta} \hat{\phi}_1(\theta) \tag{6.5.136}$$

$$\hat{\sigma}_1^2 = \frac{1}{m^2}\hat{\phi}_1(\hat{\theta}_1) \tag{6.5.137}$$

In words, $\hat{\sigma}_1^2$ is the scaled height of the highest peak of $\hat{\phi}_1(\theta)$, and $\hat{\theta}_1$ is its corresponding DOA (see (6.3.16) and (6.3.18)). As we know, the beamforming method suffers from resolution and leakage problems. However, the dominant peak of the beamforming spectrum, $\hat{\phi}_1(\theta)$, is likely to indicate that there is a source, or possibly several closely-spaced sources, at or in the vicinity of $\hat{\theta}_1$. The covariance matrix of the part of the array output due to a source signal with DOA equal to $\hat{\theta}_1$ and power equal to $\hat{\sigma}_1^2$ is given (see, e.g., (6.2.19)) by

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

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

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

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

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

where ρ is a user parameter that satisfies

$$\rho \in (0, 1] \tag{6.5.141}$$

The reason for using a value of $\rho < 1$ in (6.5.140) can be explained as follows:

(a) The assumption that there is a source with parameters $(\hat{\sigma}_1^2, \hat{\theta}_1)$ corresponding to the maximum peak of the beamforming spectrum, which led to (6.5.140), is not necessarily true. For example, there could be *several sources* clustered around $\hat{\theta}_1$ that were not resolved by the beamforming method. Subtracting only a (small) part of the beamforming response to a source signal with parameters $(\hat{\sigma}_1^2, \hat{\theta}_1)$ leaves "some power" at and around $\hat{\theta}_1$. Hence, the algorithm will likely return to this DOA region of the beamforming spectrum in future iterations, when it could have a better chance to resolve the power around $\hat{\theta}_1$ into its true constituent components.

(b) Even if there is indeed a single source at or close to $\hat{\theta}_1$, the estimation of its parameters could be affected by leakage from other sources; this leakage will be particularly strong when the source signal in question is correlated with other source signals. In such a case, (6.5.139) is a *poor estimate* of the contribution of the source in question to the beamforming spectrum. By subtracting only a part of (6.5.139) from $\hat{\phi}_1(\theta)$, we give the algorithm a chance to improve the parameter estimates of the source at or close to $\hat{\theta}_1$ in future iterations.

(c) In both situations (a) and (b), and possibly in other cases as well, in which (6.5.139) is a poor approximation of the part of the beamforming spectrum that is due to the source(s) at or around $\hat{\theta}_1$, subtracting (6.5.139) from $\hat{\phi}_1(\theta)$ fully (i.e., using $\rho = 1$) might yield a *spatial spectrum that takes on negative values* at some DOAs (as it never should). Using $\rho < 1$ in (6.5.140) reduces the likelihood that this undesirable event will happen too early in the iterative process of the CLEAN algorithm (as we discuss later).

The calculation of $\hat{\phi}_2(\theta)$, as in (6.5.140), completes the first iteration of CLEAN. In the second iteration, we proceed similarly, but using $\hat{\phi}_2(\theta)$ instead of $\hat{\phi}_1(\theta)$. Hence, we let

$$\hat{\theta}_2 = \arg \max_{\theta} \hat{\phi}_2(\theta) \tag{6.5.142}$$

$$\hat{\sigma}_2^2 = \frac{1}{m^2} \hat{\phi}_2(\hat{\theta}_2) \tag{6.5.143}$$

and

$$\hat{\phi}_3(\theta) = \hat{\phi}_2(\theta) - \rho\hat{\sigma}_2^2 \left| a^*(\theta)a(\hat{\theta}_2) \right|^2 \tag{6.5.144}$$

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

The CLEAN Algorithm

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

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

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

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

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

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

Regarding the choice of ρ in the CLEAN algorithm, there are no clear guidelines about how this choice should be made to enhance the performance of the CLEAN algorithm in a given application; $\rho \in [0.1, 0.25]$ is usually recommended. (See, e.g., [Högbom 1974; Cornwell and Bridle 1996; Schwarz 1978b].) We will make further comments on the choice of ρ later in this complement.

In the CLEAN literature, the beamforming spectral estimate $\hat{\phi}_1(\theta)$ that forms the starting point of CLEAN is called the "dirty" spectrum, in reference to its mainlobe smearing and sidelobe leakage problems. The discrete spatial spectral estimate $\{\rho\hat{\sigma}_k^2, \hat{\theta}_k\}_{k=1,2,\ldots}$ provided by the algorithm (or a suitably smoothed version of it) is called the "clean" spectrum. The iterative process that yields the "clean" spectrum is, then, called the CLEAN algorithm.

It is interesting to observe that the foregoing derivation of CLEAN is not based on a parametric model of the array output or of its covariance matrix, of the type considered in (6.2.21) or (6.4.3). More precisely, we have not made any assumption that there is a finite number of point-source signals impinging on the array, nor that the noise is spatially white. However, we have used the assumption that the covariance matrix due to a source signal has the form in (6.5.138), which cannot be true unless *the signals impinging on the array are uncorrelated with one another*. CLEAN is known to have poor performance if this parametric assumption does not hold. Hence, CLEAN is a combined nonparametric-parametric approach, which we call *semiparametric* for short.

Next, we present a *more formal derivation* of the CLEAN algorithm. Consider the following semiparametric model of the array output covariance matrix:

$$R = \sigma_1^2 a(\theta_1)a^*(\theta_1) + \sigma_2^2 a(\theta_2)a^*(\theta_2) + \cdots \tag{6.5.145}$$

As implied by the previous discussion, this is the covariance model assumed by CLEAN. Let us fit (6.5.145) to the sample covariance matrix $\hat{R}$ in a least-squares sense:

$$\min_{\{\sigma_k^2,\theta_k\}} \left\| \hat{R} - \sigma_1^2 a(\theta_1)a^*(\theta_1) - \sigma_2^2 a(\theta_2)a^*(\theta_2) - \cdots \right\|^2 \tag{6.5.146}$$

We will show that *CLEAN is a sequential algorithm for approximately minimizing the preceding LS covariance-fitting criterion.*

We begin by assuming that the initial estimates of $\sigma_2^2, \sigma_3^2, \ldots$ are equal to zero (in which case $\theta_2, \theta_3, \ldots$ are immaterial). Consequently, we obtain an estimate of the pair (σ_1^2, θ_1) by minimizing (6.5.146) with $\sigma_2^2 = \sigma_3^2 = \cdots = 0$:

$$\min_{\sigma_1^2,\theta_1} \left\| \hat{R} - \sigma_1^2 a(\theta_1)a^*(\theta_1) \right\|^2 \tag{6.5.147}$$

As shown in Complement 6.5.3, the solution to (6.5.147) is given by

$$\hat{\theta}_1 = \arg\max_{\theta} \hat{\phi}_1(\theta); \qquad \hat{\sigma}_1^2 = \frac{1}{m^2}\hat{\phi}_1(\hat{\theta}_1) \tag{6.5.148}$$

where $\hat{\phi}_1(\theta)$ is as defined previously. We reduce this power estimate by using $\rho\hat{\sigma}_1^2$ in lieu of $\hat{\sigma}_1^2$. The reasons for this reduction are discussed in points (a)–(c); in particular, we would like the residual covariance matrix $\hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1)$ to be positive definite. We will discuss this aspect in more detail after completing the derivation of CLEAN.

Next, we obtain an estimate of the pair (σ_2^2, θ_2) by minimizing (6.5.146) with $\sigma_1^2 = \rho\hat{\sigma}_1^2$, $\theta_1 = \hat{\theta}_1$ and $\sigma_3^2 = \sigma_4^2 = \cdots = 0$:

$$\min_{\sigma_2^2,\theta_2} \left\| \hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1) - \sigma_2^2 a(\theta_2)a^*(\theta_2) \right\|^2 \tag{6.5.149}$$

The solution to (6.5.149) can be shown to be (as in solving (6.5.147))

$$\hat{\theta}_2 = \arg\max_{\theta} \hat{\phi}_2(\theta); \qquad \hat{\sigma}_2^2 = \frac{1}{m^2}\hat{\phi}_2(\hat{\theta}_2) \tag{6.5.150}$$

where

$$\begin{aligned} \hat{\phi}_2(\theta) &= a^*(\theta)\left[\hat{R} - \rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1)\right]a(\theta) \\ &= \hat{\phi}_1(\theta) - \rho\hat{\sigma}_1^2 \left|a^*(\theta)a(\hat{\theta}_1)\right|^2 \end{aligned} \tag{6.5.151}$$

Observe that (6.5.148) and (6.5.150) coincide with (6.5.136)–(6.5.137) and (6.5.142)–(6.5.143). Evidently, continuing the previous iterative process, for which (6.5.148) and (6.5.150) are the first two steps, leads to the CLEAN algorithm on page 328.

The foregoing derivation of CLEAN sheds some light on the properties of this algorithm. First, note that *the LS covariance-fitting criterion in (6.5.146) is decreased at each iteration of CLEAN*. For instance, consider the first iteration. A straightforward calculation shows that

$$\begin{aligned}\left\|\hat{R}-\rho\hat{\sigma}_1^2 a(\hat{\theta}_1)a^*(\hat{\theta}_1)\right\|^2 &= \|\hat{R}\|^2-2\rho\hat{\sigma}_1^2 a^*(\hat{\theta}_1)\hat{R}a(\hat{\theta}_1)+m^2\rho^2\hat{\sigma}_1^4 \\ &= \|\hat{R}\|^2-\rho(2-\rho)m^2\hat{\sigma}_1^4 \end{aligned} \tag{6.5.152}$$

Clearly, (6.5.152) is less than $\|\hat{R}\|^2$ for any $\rho \in (0,2)$, and the maximum decrease occurs for $\rho = 1$ (as expected). A similar calculation shows that the criterion in (6.5.146) monotonically decreases as we continue the iterative process, for any $\rho \in (0,2)$, and that at each iteration the maximum decrease occurs for $\rho = 1$. As a consequence, we might think of choosing $\rho = 1$, but this is not advisable. The reason is that our goal is not only to decrease the fitting criterion (6.5.146) as much and as fast as possible, but also to ensure that the residual covariance matrices

$$\hat{R}_{k+1} = \hat{R}_k - \rho\hat{\sigma}_k^2 a(\hat{\theta}_k)a^*(\hat{\theta}_k); \qquad \hat{R}_1 = \hat{R} \tag{6.5.153}$$

remain positive definite for $k = 1, 2, \ldots$; otherwise, fitting $\sigma_{k+1}^2 a(\theta_{k+1})a^*(\theta_{k+1})$ to $\hat{R}_{k+1}$ would make little statistical sense. By a calculation similar to that in equation (6.5.33) of Complement 6.5.3, it can be shown that the condition $\hat{R}_{k+1} > 0$ is equivalent to

$$\rho < \frac{1}{\hat{\sigma}_k^2 a^*(\hat{\theta}_k)\hat{R}_k^{-1}a(\hat{\theta}_k)} \tag{6.5.154}$$

Note that the right-hand side of (6.5.154) is bounded above by 1, because, by the Cauchy–Schwartz inequality,

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

Also note that, depending on the scenario under consideration, satisfaction of the inequality in (6.5.154) for $k = 1, 2, \ldots$ could require choosing a value for ρ much less than 1. To summarize, the preceding discussion has provided a *a precise argument for choosing* $\rho < 1$ (or even $\rho \ll 1$) in the CLEAN algorithm.

The LS covariance-fitting derivation of CLEAN also makes *the semiparametric nature of CLEAN* more transparent. Specifically, the discussion has shown that CLEAN fits the semiparametric covariance model in (6.5.145) to the sample covariance matrix $\hat{R}$.

Finally, note that, although there is a significant literature on CLEAN, its statistical properties are not well understood; in fact, other than the preliminary study of CLEAN reported in [SCHWARZ 1978B], there appear to be very few statistical studies in the literature. The derivation of CLEAN based on the LS covariance-fitting criterion in (6.5.146) could also be useful in *understanding the statistical properties of CLEAN*. However, we will not attempt to provide a statistical analysis of CLEAN in this complement.

6.5.8 Unstructured and Persymmetric ML Estimates of the Covariance Matrix

Let $\{y(t)\}_{t=1,2,\ldots}$ 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

$$p\big(y(t)\big) = \frac{1}{\pi^m |R|} e^{-y^*(t)R^{-1}y(t)} \tag{6.5.155}$$

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

$$\{y(1), \ldots, y(N)\} \tag{6.5.156}$$

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 (6.5.156) is given by

$$\begin{aligned} p\big(y(1), \ldots, y(N)\big) &= \prod_{t=1}^{N} p\big(y(t)\big) \\ &= \frac{1}{\pi^{mN} |R|^N} e^{-\sum_{t=1}^{N} y^*(t)R^{-1}y(t)} \end{aligned} \tag{6.5.157}$$

The maximum likelihood (ML) estimate of the covariance matrix R, based on the sample in (6.5.156), is given by the maximizer of the likelihood function in (6.5.157) (see Section B.1 in Appendix B) or, equivalently, by the minimizer of the negative log-likelihood function:

$$-\ln p\big(y(1), \ldots, y(N)\big) = mN \ln(\pi) + N \ln |R| + \sum_{t=1}^{N} y^*(t)R^{-1}y(t) \tag{6.5.158}$$

The part of (6.5.158) that depends on R is given (after multiplication by $\frac{1}{N}$) by

$$\ln|R| + \frac{1}{N}\sum_{t=1}^{N} y^*(t)R^{-1}y(t) = \ln|R| + \operatorname{tr}\left(R^{-1}\hat{R}\right) \tag{6.5.159}$$

where

$$\hat{R} = \frac{1}{N}\sum_{t=1}^{N} y(t)y^*(t) \qquad (m \times m) \tag{6.5.160}$$

In this complement, we discuss the minimization of (6.5.159) with respect to R, which yields the ML estimate of R, under either of the following two assumptions:

A: R has no assumed structure

or

B: R is persymmetric.

As explained in Section 4.8, R is persymmetric (or centrosymmetric) if and only if

$$JR^TJ = R \quad \Longleftrightarrow \quad R = \frac{1}{2}\left(R + JR^TJ\right) \tag{6.5.161}$$

where J is the so-called reversal matrix defined in (4.8.4).

Remark: If $y(t)$ is the output of an array that is uniform and linear and the source signals are uncorrelated with one another, then the covariance matrix R is Toeplitz, and, hence, persymmetric. ■

We will show that the unstructured ML estimate of R, denoted $\hat{R}_{U,ML}$, is given by the standard sample covariance matrix in (6.5.160),

$$\boxed{\hat{R}_{U,ML} = \hat{R}} \tag{6.5.162}$$

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

$$\boxed{\hat{R}_{P,ML} = \frac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right)} \tag{6.5.163}$$

To prove (6.5.162), we need to show (see (6.5.159)) that

$$\ln|R| + \operatorname{tr}\left(R^{-1}\hat{R}\right) \geq \ln|\hat{R}| + m \quad \text{for any } R > 0 \tag{6.5.164}$$

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

$$\operatorname{tr}\left(R^{-1}\hat{R}\right) = \operatorname{tr}\left(R^{-1}\hat{C}\hat{C}^*\right) = \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) \tag{6.5.165}$$

Using (6.5.165) in (6.5.164), we obtain the series of equivalences

$$\begin{aligned}(6.5.164) &\iff \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) - \ln\left|R^{-1}\hat{R}\right| \geq m \\ &\iff \operatorname{tr}\left(\hat{C}^*R^{-1}\hat{C}\right) - \ln\left|\hat{C}^*R^{-1}\hat{C}\right| \geq m \\ &\iff \sum_{k=1}^{m}(\lambda_k - \ln\lambda_k - 1) \geq 0 \end{aligned} \tag{6.5.166}$$

where $\{\lambda_k\}$ are the eigenvalues of the matrix $\hat{C}^*R^{-1}\hat{C}$.

Next, with reference to (6.5.166), we show that

$$f(\lambda) \triangleq \lambda - \ln\lambda - 1 \geq 0 \quad \text{for any } \lambda > 0 \tag{6.5.167}$$

To verify (6.5.167), observe that

$$f'(\lambda) = 1 - \frac{1}{\lambda}; \qquad f''(\lambda) = \frac{1}{\lambda^2}$$

Hence, the function $f(\lambda)$ in (6.5.167) has a unique minimum at $\lambda = 1$, and $f(1) = 0$; this proves (6.5.167). With this observation, the proof of (6.5.166), and therefore of (6.5.162), is complete.

The proof of (6.5.163) is even simpler. In view of (6.5.161), we have that

$$\begin{aligned}\operatorname{tr}\left(R^{-1}\hat{R}\right) &= \operatorname{tr}\left[\left(JR^TJ\right)^{-1}\hat{R}\right] = \operatorname{tr}\left(R^{-T}J\hat{R}J\right) \\ &= \operatorname{tr}\left(R^{-1}J\hat{R}^TJ\right)\end{aligned} \tag{6.5.168}$$

Hence, the function to be minimized with respect to R (under the constraint (6.5.161)) can be written as

$$\ln|R| + \operatorname{tr}\left[R^{-1}\cdot\tfrac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right)\right] \tag{6.5.169}$$

As was shown earlier in this complement, the unstructured minimizer of (6.5.169) is given by

$$R = \frac{1}{2}\left(\hat{R} + J\hat{R}^TJ\right) \tag{6.5.170}$$

Because (6.5.170) satisfies the persymmetry constraint, by construction, it also gives the constrained minimizer of the negative log-likelihood function; hence, the proof of (6.5.163) is concluded as well.

The reader interested in more details on the topic of this complement, including a comparison of the statistical estimation errors associated with $\hat{R}_{U,ML}$ and $\hat{R}_{P,ML}$, can consult [Jansson and Stoica 1999].

6.6 EXERCISES

Exercise 6.1: Source Localization by Using a Sensor in Motion
This exercise illustrates how the directions of arrival of planar waves can be found by using a single moving sensor. Conceptually, this problem is related to that of DOA estimation by sensor-array methods. Indeed, we can think of a sensor in motion as creating a *synthetic aperture* similar to the one corresponding to a physical array of spatially distributed sensors.

Assume that the sensor has a linear motion with constant speed equal to v. Also, assume that the sources are far-field point emitters at fixed locations in the same plane as the sensor. Let θ_k denote the kth DOA parameter (defined as the angle between the direction of wave propagation and the normal to the sensor trajectory). Finally, assume that the sources emit sinusoidal signals $\{\alpha_k e^{i\omega t}\}_{k=1}^n$ with the same (center) frequency ω. (These signals could be reflections of a probing sinusoidal signal from different point scatterers of a target, in which case it is not restrictive to assume that they all have the same frequency.)

Show that, under the previous assumptions and after elimination of the high-frequency component corresponding to the frequency ω, the sensor output signal can be written as

$$s(t) = \sum_{k=1}^{n} \alpha_k e^{i\omega_k^D t} + e(t) \tag{6.6.1}$$

where $e(t)$ is measurement noise and ω_k^D is the kth *Doppler frequency*, defined by

$$\omega_k^D = -\frac{v \cdot \omega}{c} \sin\theta_k$$

with c denoting the velocity of signal propagation. Conclude from (6.6.1) that the DOA-estimation problem associated with the scenario under consideration can be solved by using the estimation methods discussed in this chapter and in Chapter 4 (provided that the sensor speed v can be accurately determined).

Exercise 6.2: Beamforming Resolution for Uniform Linear Arrays
Consider a ULA comprising m sensors, with interelement spacing equal to d. Let λ denote the wavelength of the signals impinging on the array. According to the discussion in Chapter 2, the *spatial-frequency resolution* of the beamforming used with this ULA is given by

$$\Delta\omega_s = \frac{2\pi}{m} \quad \Longleftrightarrow \quad \Delta f_s = \frac{1}{m} \tag{6.6.2}$$

Make use of the previous observation to show that the *DOA resolution* of beamforming for signals coming from *broadside* is

$$\boxed{\Delta\theta \simeq \sin^{-1}(1/L)} \tag{6.6.3}$$

where L is the array's length measured in wavelengths:

$$L = \frac{(m-1)d}{\lambda} \tag{6.6.4}$$

Explain how (6.6.3) approximately reduces to (6.3.20), for sufficiently large L.

Next, show that, for signals impinging from an *arbitrary direction angle* θ, the *DOA resolution* of beamforming is approximately

$$\Delta\theta \simeq \frac{1}{L|\cos\theta|} \tag{6.6.5}$$

Hence, for signals coming from nearly end-fire directions, the DOA resolution is much worse than what is suggested in (6.3.20).

Exercise 6.3: Beamforming Resolution for Arbitrary Arrays
The *beampattern*

$$W(\theta) = |a^*(\theta)a(\theta_0)|^2, \qquad (\text{some } \theta_0)$$

has the same shape as a spectral window: It has a peak at $\theta = \theta_0$, is symmetric about that point, and the peak is narrow (for large enough values of m). Consequently, the *beamwidth* of the array with direction vector $a(\theta)$ can approximately be derived by using the window bandwidth formula proven in Exercise 2.15:

$$\Delta\theta \simeq 2\sqrt{|W(\theta_0)/W''(\theta_0)|} \tag{6.6.6}$$

Now, the array's beamwidth and the resolution of beamforming are closely related. To see this, consider the case where the array output covariance matrix is given by (6.4.3). Let $n = 2$, and assume that $P = I$ (for simplicity of explanation). The average beamforming spectral function is then given by

$$a^*(\theta)Ra(\theta) = |a^*(\theta)a(\theta_1)|^2 + |a^*(\theta)a(\theta_2)|^2 + m\sigma^2$$

which clearly shows that the sources with DOAs θ_1 and θ_2 are resolvable by beamforming if and only if $|\theta_1 - \theta_2|$ is larger than the array's beamwidth. Consequently, we can approximately determine the beamforming resolution by using (6.6.6). Specialize equation (6.6.6) to a ULA, and compare with the results obtained in Exercise 6.2.

Exercise 6.4: Beamforming Resolution for L-Shaped Arrays
Consider an m-element array, with m odd, shaped as an "L" with element spacing d. Thus, the array elements are located at points $(0,0)$, $(0,d), \ldots, (0, d(m-1)/2)$ and $(d,0), \ldots, (d(m-$

$1)/2, 0)$. Using the results in Exercise 6.3, find the DOA resolution of beamforming for signals coming from an angle θ. What are the minimum and maximum resolution, and for what angles are these extremal resolutions realized? Compare your results with the m-element ULA case in Exercise 6.2.

Exercise 6.5: Relationship between Beamwidth and Array-Element Locations
Consider an m-element planar array with elements located at $r_k = [x_k, y_k]^T$ for $k = 1, \ldots, m$. Assume that the array is centered at the origin, so $\sum_{k=1}^{m} r_k = 0$. Use equation (6.6.6) to show that the array beamwidth at direction θ_0 is given by

$$\Delta\theta \simeq \sqrt{2}\frac{\lambda}{2\pi}\frac{1}{D(\theta_0)} \tag{6.6.7}$$

where $D(\theta_0)$ is the root-mean-square distance of the array elements to the origin in the direction orthogonal to θ_0 (see Figure 6.8):

$$D(\theta_0) = \sqrt{\frac{1}{m}\sum_{k=1}^{m} d_k^2(\theta_0)}, \qquad d_k(\theta_0) = x_k \sin\theta_0 - y_k \cos\theta_0$$

As in Exercise 2.15, the beamwidth approximation in equation (6.6.7) slightly underestimates the true beamwidth; a better approximation is given by

$$\Delta\theta \simeq 1.15\sqrt{2}\frac{\lambda}{2\pi}\frac{1}{D(\theta_0)} \tag{6.6.8}$$

Exercise 6.6: Isotropic Arrays
An array whose beamwidth is the same for all directions is said to be *isotropic*. Consider an m-element planar array with elements located at $r_k = [x_k, y_k]^T$ for $k = 1, \ldots, m$ and centered at

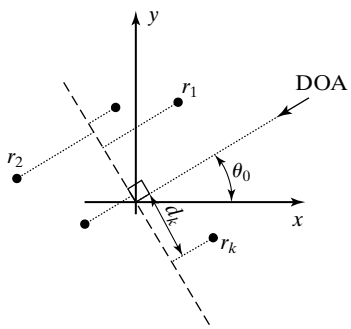


Figure 6.8 Array-element projected distances from the origin for DOA angle θ_0 (see Exercise 6.5).

the origin ($\sum_{k=1}^{m} r_k = 0$), as in Exercise 6.5. Show that the array beamwidth (as given by (6.6.7)) is the same for all DOAs if and only if

$$R^T R = cI_2 \tag{6.6.9}$$

where

$$R = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix}$$

and where c is a positive constant. (See [Baysal and Moses 2003] for additional details and properties of isotropic arrays.)

Exercise 6.7: Grating Lobes

The results of Exercise 6.2 might suggest that an m-element ULA can have very high resolution simply by using a large array element spacing d. However, there is an ambiguity associated with choosing $d > \lambda/2$; this drawback is sometimes referred to as the problem of *grating lobes*. Identify this drawback, and discuss what ambiguities exist as a function of d. (Refer to the discussion on ULAs in Section 6.2.2.)

One potential remedy to this drawback is to use two ULAs: one with m_1 elements and element spacing $d_1 = \lambda/2$, and another with m_2 elements and element spacing d_2. Discuss how to choose m_1, m_2, and d_2 to both avoid ambiguities and increase resolution over a conventional ULA with element spacing $d = \lambda/2$ and $m_1 + m_2$ elements. Consider as an example using a 10-element ULA with $d_2 = 3\lambda/2$ for the second ULA; find m_1 to resolve ambiguities in this array. Finally, discuss any potential drawbacks of the two-array approach.

Exercise 6.8: Beamspace Processing

Consider an array comprising many sensors ($m \gg 1$). Such an array should be able to resolve sources that are quite closely spaced (*cf.* (6.3.20) and the discussion in Exercise 6.3). There is, however, a price to be paid for the high-resolution performance achieved by using many sensors: the computational burden associated with the *elementspace processing* (ESP) (i.e., the direct processing of the output of all sensors) may be prohibitively high, and the involved circuitry (A–D converters, etc.) could be quite expensive.

Let B^* be an $\bar{m} \times m$ matrix with $\bar{m} < m$, and consider the transformed output vector $B^* y(t)$. The latter vector satisfies the following equation (*cf.* (6.2.21)):

$$B^* y(t) = B^* A s(t) + B^* e(t) \tag{6.6.10}$$

The transformation matrix B^* can be interpreted as a *beamformer* or *spatial filter* acting on $y(t)$. Estimation of the DOAs of the signals impinging on the array using $B^* y(t)$ is called *beamspace processing* (BSP). Since $\bar{m} < m$, BSP should have a lower computational burden than ESP. The critical question is then how to choose the beamformer B so as not to significantly degrade the performance achievable by ESP.

Assume that a certain DOA sector is known to contain the source(s) of interest (whose DOAs are designated by the generic variable θ_0). Using this information, design a matrix B^* that passes the signals from direction θ_0 approximately undistorted. Choose B in such a way that the noise in beamspace, $B^*e(t)$, is still spatially white. For a given sector size, discuss the tradeoff between the computational burden associated with BSP and the distorting effect of the beamformer on the desired signals. Finally, use the results of Exercise 6.3 to show that the resolution of beamforming in elementspace and beamspace are nearly the same, under the previous conditions.

Exercise 6.9: Beamspace Processing (cont'd)
In this exercise, for simplicity, we consider the beamspace processing (BSP) equation (6.6.10) for the case of a single source ($n = 1$):

$$B^*y(t) = B^*a(\theta)s(t) + B^*e(t) \tag{6.6.11}$$

The elementspace processing (ESP) counterpart of (6.6.11) (*cf.* (6.2.19)) is

$$y(t) = a(\theta)s(t) + e(t) \tag{6.6.12}$$

Assume that $\|a(\theta)\|^2 = m$ (see (6.3.11)) and that the $\bar{m} \times m$ matrix B^* is unitary (i.e., $B^*B = I$). Furthermore, assume that

$$a(\theta) \in \mathcal{R}(B) \tag{6.6.13}$$

To satisfy (6.6.13), we need knowledge about a DOA sector that contains θ, which is usually assumed to be available in BSP applications; note that the narrower this sector, the smaller the value we can choose for $\bar{m}$. As $\bar{m}$ decreases, the implementation advantages of BSP compared with ESP become more significant. However, the DOA estimation performance achievable by BSP might be expected to decrease as $\bar{m}$ decreases. As indicated in Exercise 6.8, this is not necessarily the case. In the present exercise, we lend further support to the fact that the estimation performances of ESP and BSP can be quite similar to one another, provided that condition (6.6.13) is satisfied. To be specific, define the array SNR for (6.6.12) as

$$\frac{E\left\{\|a(\theta)s(t)\|^2\right\}}{E\|e(t)\|^2} = \frac{mP}{m\sigma^2} = \frac{P}{\sigma^2} \tag{6.6.14}$$

where P denotes the power of $s(t)$. Show that the "array SNR" for the BSP equation, (6.6.11), is $m/\bar{m}$ times that in (6.6.14). Conclude that this increase in the array SNR associated with BSP might counterbalance the presumably negative impact on DOA performance due to the decrease from m to $\bar{m}$ in the number of observed output signals.

Exercise 6.10: Beamforming and MUSIC under the Same Umbrella
Define the scalars

$$Y_t^*(\theta) = a^*(\theta)y(t), \qquad t = 1, \ldots, N.$$

By using previous notation, we can write the beamforming spatial spectrum in (6.3.18) as

$$Y^*(\theta)WY(\theta) \tag{6.6.15}$$

where

$$W = (1/N)I \qquad \text{(for beamforming)}$$

and

$$Y(\theta) = [Y_1(\theta) \dots Y_N(\theta)]^T$$

Show that the MUSIC spatial pseudospectrum

$$a^*(\theta)\hat{S}\hat{S}^*a(\theta) \tag{6.6.16}$$

(see Sections 4.5 and 6.4.3) can also be put in the form (6.6.15), for a certain "weighting matrix" W. The columns of the matrix $\hat{S}$ in (6.6.16) are the n principal eigenvectors of the sample covariance matrix $\hat{R}$ in (6.3.17).

Exercise 6.11: Subspace Fitting Interpretation of MUSIC
In words, the result (4.5.9) (on which MUSIC for both frequency and DOA estimation is based) says that the direction vectors $\{a(\theta_k)\}$ belong to the subspace spanned by the columns of S. Therefore, we can think of estimating the DOAs by choosing θ (a generic DOA variable) so that the distance between $a(\theta)$ and the closest vector in the span of $\hat{S}$ is minimized—that is,

$$\min_{\beta,\theta} \|a(\theta) - \hat{S}\beta\|^2 \tag{6.6.17}$$

where $\|\cdot\|$ denotes the Euclidean vector norm. Note that the dummy vector variable β in (6.6.17) is defined in such a way that $\hat{S}\beta$ is closest to $a(\theta)$ in Euclidean norm.

Show that the DOA estimation method derived from the subspace-fitting criterion (6.6.17) is the same as MUSIC.

Exercise 6.12: Subspace Fitting Interpretation of MUSIC (cont'd)
The result (4.5.9) can also be invoked to arrive at the following subspace fitting criterion:

$$\min_{B,\theta} \|A(\theta) - \hat{S}B\|_F^2 \tag{6.6.18}$$

where $\|\cdot\|_F$ stands for the Frobenius matrix norm and θ is now the vector of all DOA parameters. This criterion seems to be a more general version of equation (6.6.17) in Exercise 6.11. Show that the minimization of the multidimensional subspace fitting criterion in (6.6.18), with respect to the DOA vector θ, still leads to the one-dimensional MUSIC method. **Hint:** It will be useful to refer to the type of result proven in equations (4.3.12)–(4.3.16) in Section 4.3.

Exercise 6.13: Subspace Fitting Interpretation of MUSIC (cont'd)
The subspace fitting interpretations of the previous two exercises provide some insights into the properties of the MUSIC estimator. Assume, for instance, that two or more source signals are coherent. Make use of the subspace fitting interpretation in Exercise 6.12 to show that MUSIC cannot be expected to yield meaningful results in such a case. Follow the line of your argument, explaining why MUSIC fails in the case of coherent signals, to suggest a subspace fitting criterion that works in such a case. Discuss the computational complexity of the method based on the latter criterion.

Exercise 6.14: Modified MUSIC for Coherent Signals
Consider an m-element ULA. Assume that n signals impinge on the array at angles $\{\theta_k\}_{k=1}^n$ and, also, that some signals are coherent (so that the signal covariance matrix P is singular). Derive a modified MUSIC DOA estimator for this case, analogous to the modified MUSIC frequency estimator in Section 4.5, and show that this method is capable of estimating the n DOAs even in the coherent-signal case.

COMPUTER EXERCISES

Tools for Array Signal Processing:
The text website `www.prenhall.com/stoica` contains the following MATLAB functions for use in DOA estimation:

- `Y=uladata(theta,P,N,sig2,m,d)`
  Generates an $m \times N$ data matrix $Y = [y(1), \ldots, y(N)]$ for a ULA with `n` sources arriving at angles (in degrees from -90° to 90°) given by the elements of the $n \times 1$ vector `theta`. The source signals are zero-mean Gaussian with covariance matrix $P = E\{s(t)s^*(t)\}$. The noise component is spatially white Gaussian with covariance $\sigma^2 I$, where $\sigma^2 =$`sig2`. The element spacing is equal to `d` in wavelengths.
- `phi=beamform(Y,L,d)`
  Implements the beamforming spatial spectral estimate in equation (6.3.18) for an m-element ULA with sensor spacing `d` in wavelengths. The $m \times N$ matrix `Y` is as defined above. The parameter `L` controls the DOA sampling, and `phi` is the spatial spectral estimate `phi`$= [\hat{\phi}(\theta_1), \ldots, \hat{\phi}(\theta_L)]$, where $\theta_k = -\frac{\pi}{2} + \frac{\pi k}{L}$.
- `phi=capon_sp(Y,L,d)`
  Implements the Capon spatial spectral estimator in equation (6.3.26); the input and output parameters are defined as those in `beamform`.
- `theta=root_music_doa(Y,n,d)`
  Implements the Root MUSIC method in Section 4.5, adapted for spatial spectral estimation using a ULA. The parameters `Y` and `d` are as in `beamform`, and `theta` is the vector containing the `n` DOA estimates $[\hat{\theta}_1, \ldots, \hat{\theta}_n]^T$.
- `theta=esprit_doa(Y,n,d)`
  Implements the ESPRIT method for a ULA. The parameters `Y` and `d` are as in `beamform`, and `theta` is the vector containing the `n` DOA estimates $[\hat{\theta}_1, \ldots, \hat{\theta}_n]^T$. The two subarrays for ESPRIT are made from the first $m-1$ and last $m-1$ elements of the array.

Exercise C6.15: Comparison of Spatial Spectral Estimators
Simulate the following scenario: Two signals with wavelength λ impinge on an array of sensors from DOAs $\theta_1 = 0^\circ$ and a θ_2 that will be varied. The signals are mutually uncorrelated complex Gaussian with unit power, so that $P = E\{s(t)s^*(t)\} = I$. The array is a 10-element ULA with element spacing $d = \lambda/2$. The measurements are corrupted by additive complex Gaussian white noise with unit power. $N = 100$ snapshots are collected.

(a) Let $\theta_2 = 15^\circ$. Compare the results of the beamforming, Capon, Root MUSIC, and ESPRIT methods for this example. The results can be shown by plotting the spatial spectrum estimates from beamforming and Capon for 50 Monte Carlo experiments; for Root MUSIC and ESPRIT, plot vertical lines of equal height located at the DOA estimates from the 50 Monte Carlo experiments. How do the methods compare? Are the properties of the various estimators analogous to the time-series case for two sinusoids in noise?
(b) Repeat for $\theta_2 = 7.5^\circ$.

Exercise C6.16: Performance of Spatial Spectral Estimators for Coherent Source Signals
In this exercise, we will see what happens when the source signals are fully correlated (or coherent). Use the same parameters and estimation methods as in Exercise C6.15 with $\theta_2 = 15^\circ$, but with

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Note that the sources are coherent as $\text{rank}(P) = 1$.

Compare the results of the four methods for this case, again by plotting the spatial spectrum and "DOA line spectrum" estimates (as in Exercise C6.15) for 50 Monte Carlo experiments from each estimator. Which method appears to be the best in this case?

Exercise C6.17: Spatial Spectral Estimators Applied to Measured Data
Apply the four DOA estimators from Exercise C6.15 to the real data in the file `submarine.mat`, which can be found at the text website `www.prenhall.com/stoica`. These data are underwater measurements collected by the Swedish Defense Agency in the Baltic Sea. The 6-element array of hydrophones used in the experiment can be assumed to be a ULA with inter-element spacing equal to 0.9 m. The wavelength of the signal is approximately 5.32 m. Can you find the "submarine(s)?"