

# Chapter 7

# Adaptive Beamformers

## 7.1 Introduction

In this chapter we develop techniques for implementing the algorithms developed in Chapter 6 using the incoming data. In Chapter 6, we assumed that we knew the signal directions and the various spatial spectral matrices ( $\mathbf{S}_x$  or  $\mathbf{S}_n$ ). In actual applications, we must estimate these quantities (or appropriate surrogates) from the incoming data. The resulting beamformers will adapt to the incoming data and are referred to as **adaptive beamformers**. In this chapter we develop the theory and practice of adaptive beamformers.

The adaptive beamformers that we develop can be divided into three general categories:

- (i) Beamformers that estimate the spatial spectral matrix  $\mathbf{S}_x$  or  $\mathbf{S}_n$  (or the correlation  $\mathbf{R}_x$  or  $\mathbf{R}_n$  if the implementation is in the time domain) and use the estimate in the appropriate formula from Chapter 6 (e.g., (6.14), (6.71)). This implementation requires the inversion of the sample covariance matrix and is frequently referred to as the sample matrix inversion (SMI) technique. It is also referred to as the direct matrix inversion (DMI) technique (e.g., [MM80]) or the estimate-and-plug technique (e.g., [Hay85]). It is a block data processor.
- (ii) Beamformers that implement the inversion recursively are the second category. We reformulate the algorithm as a least squares algorithm and develop a recursive version. These recursive least squares (RLS) implementations potentially have performance that is similar to the SMI beamformer.
- (iii) A third approach is to adapt classical steepest descent algorithms to

the optimization problem in order to find  $\mathbf{w}_{opt}$ . This approach leads to the least mean square (LMS) algorithm. These algorithms require less computation, but converge slower to the optimum solution.

In Section 7.2, we discuss various techniques for estimating the covariance matrix (or spectral matrix) of the array output. The eigenvalues and eigenvectors play a key role in several subsequent discussions. We show how they can be derived directly from the data using a singular value decomposition (SVD).

In Section 7.3, we implement the MVDR and MPDR beamformers using a technique called **sample matrix inversion** (SMI). As the name implies, we use  $\mathbf{C}_x$ , the sample correlation matrix in place of  $\mathbf{S}_x$  and invert it to obtain the MVDR or MPDR beamformers. We show how the use of a finite amount of data affects the performance.

In Section 7.4, we reformulate the problem using a least squares formulation and introduce an exponentially weighted sample spectral matrix  $\Phi$ . We find that the resulting MPDR beamformer is identical to the MPDR beamformer of Chapter 6, with the ensemble spatial spectral matrix  $\mathbf{S}_x$  replaced by  $\Phi$ . We then develop a recursive implementation of the algorithm which is denoted as the **recursive least squares** (RLS) algorithm. We compare its performance to the SMI algorithm and develop various diagonal loading methods.

In Section 7.5, we develop more efficient recursive algorithms that have better numerical stability and are computationally simpler than the algorithms in Section 7.4. We show how these algorithms can be put in a structure that can be implemented efficiently in VLSI.

In Section 7.6, we begin discussion of a group of algorithms that rely on the quadratic characteristic of the error surface and utilize gradient techniques to find the optimum weight vector. Section 7.6 discusses steepest descent algorithms. These algorithms are deterministic and provide background for the stochastic gradient algorithms that we actually employ.

In Section 7.7, we develop **least mean-square** (LMS) algorithms and investigate their performance. The LMS algorithms are computationally much simpler than the SMI and RLS algorithms, but they converge much more slowly to the optimum solution.

In Section 7.8, we study the important problem of detecting the number of plane-wave signals (including both desired signals and interfering signals) that are impinging on the array. In this chapter, we need this information in order to implement the adaptive version of the eigenspace beamformers that we developed in Section 6.8. In Chapters 8 and 9, we will need this

information for parameter estimation.

In Section 7.9, we study adaptive eigenvector beamformers. We first utilize an SMI implementation and find that we obtain faster convergence to the optimum beamformer because of the reduced degrees of freedom.

In Section 7.10, we study beamspace adaptive beamformers. In Section 6.9, we found that (in most scenarios) we could achieve performance similar to element-space beamformers with reduced computational complexity. In the adaptive case, we find that we can obtain faster convergence to the optimum solution because of the reduced dimension.

In Section 7.11, we study the adaptive implementation of the broadband beamformers that we developed in Section 6.13. We restrict our discussion to time-domain implementations.

In Section 7.12, we summarize our results and discuss some open issues. The structure of the chapter is shown in Table 7.1.

## 7.2 Estimation of Spatial Spectral Matrices

In the discussion up to this point in the book, we have assumed that the second-order statistics of the input process were known. In practice, we usually have to estimate these statistics from a finite amount of data. We have available a sequence of snapshots,  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K$ , where  $\mathbf{X}_k$  is an  $N$ -dimensional vector corresponding to the frequency-domain snapshot at time  $k$ . We process these snapshots to obtain an estimate of  $\mathbf{S}_x$ , which we denote as  $\hat{\mathbf{S}}_x$ .

The two issues of interest are:

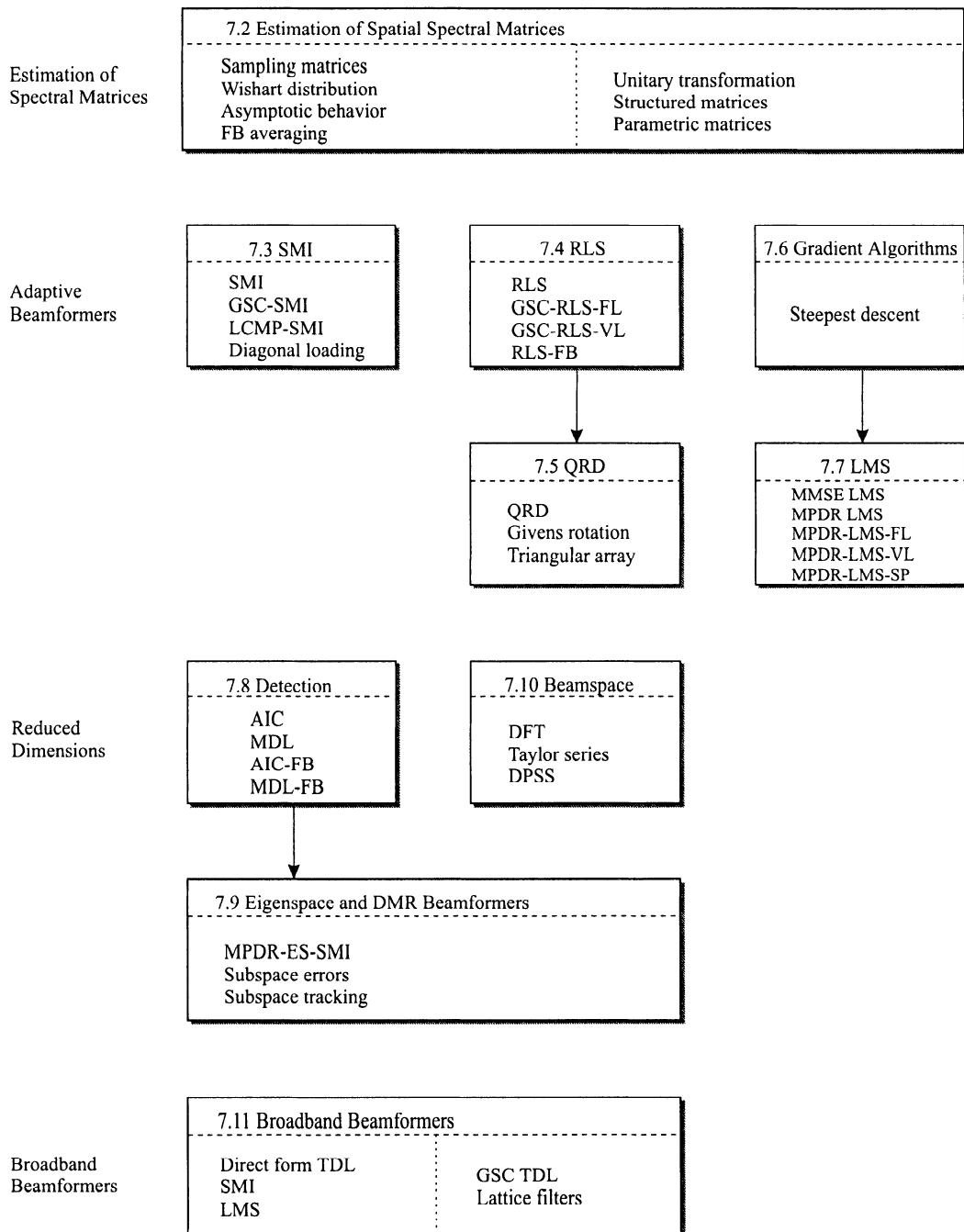
- (i) What is the appropriate estimator?
- (ii) How well does it perform?

We will find that, as we assume more prior information about the process, the estimators become more accurate. However, they also become more computationally complex and dependent on the prior assumptions. A particular problem of interest to us is the case in which  $\mathbf{S}_x$  has the structure in (6.173) but  $\mathbf{C}_x$ , the sample spectral matrix defined in (7.3), does not have that structure. Thus, our minimization depends on how much structure we impose on  $\hat{\mathbf{S}}_x$ .

A logical measure of estimator performance is the Frobenius norm of the error matrix,

$$\xi_F = \| \hat{\mathbf{S}}_x - \mathbf{S}_x \|_F . \quad (7.1)$$

Table 7.1 Structure of Chapter 7



However, we find that the appropriate measure of the quality of the estimate depends on how we are going to use it and that choosing an estimator to minimize  $\xi_F$  may not be the best approach.

In this section, we develop several methods for estimating  $\mathbf{S}_x$  and its eigendecomposition.

In Section 7.2.1, we utilize the sample spectral matrix as an estimator. It applies to arbitrary array geometries and is the mostly commonly used technique.

In Section 7.2.2, we discuss the asymptotic behavior (large  $K$ ) of the eigenvalues and eigenvectors of the sample spectral matrix.

In Section 7.2.3, we restrict our attention to arrays with conjugate symmetric array manifolds and derive an estimator that uses FB averaging of the data. This leads to improved performance and computational advantages.

In Section 7.2.4, we consider the case of uniform linear arrays. If the input signals are uncorrelated, then the ensemble spectral matrix is Toeplitz. In this section, we discuss an estimate of the spectral matrix subject to the Toeplitz constraint.

In Section 7.2.5, we introduce the idea of parametric spectral estimation, but do not pursue it at this point.

In Section 7.2.6, we review the singular value decomposition (SVD). This is a technique in which we operate directly on the data matrix to obtain the eigenvalues and eigenvectors of the sample spectral matrix. This technique is widely used in practice as an alternative to conventional eigendecomposition.

In Section 7.2.7, we summarize our results.

### 7.2.1 Sample Spectral Matrices

In this section we discuss the use of the sample spectral matrix as an estimate of the spectral matrix  $\mathbf{S}_x$ .

We have available a sequence of snapshots,  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K$ , where  $\mathbf{X}_k$  is an  $N$ -dimensional vector corresponding to the frequency-domain snapshot at time  $k$ . The snapshots are modeled as statistically independent, identically distributed, complex Gaussian random vectors.

The joint probability density is

$$p_{\mathbf{X}}(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K) = \prod_{i=1}^K \frac{\exp \left[ -\mathbf{X}_k^H \mathbf{S}_x^{-1} \mathbf{X}_k \right]}{\pi \det [\mathbf{S}_x]}. \quad (7.2)$$

We define the sample spectral matrix  $\mathbf{C}_x$  as<sup>1</sup>

$$\mathbf{C}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{X}_k \mathbf{X}_k^H = \frac{1}{K} \sum_{k=1}^K \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.3)$$

We can also write  $\mathbf{C}_x$  in terms of a  $N \times K$  data matrix,  $\tilde{\mathbf{X}}$ ,

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{K}} \left[ \begin{array}{c|c|c|c} \mathbf{X}(1) & \mathbf{X}(2) & \cdots & \mathbf{X}(K) \end{array} \right], \quad (7.4)$$

or

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{K}} \left[ \begin{array}{c|c|c|c} X_0(1) & X_0(2) & \cdots & X_0(K) \\ X_1(1) & X_1(2) & \cdots & X_1(K) \\ \vdots & & \ddots & \\ X_{N-1}(1) & X_{N-1}(2) & \cdots & X_{N-1}(K) \end{array} \right], \quad (7.5)$$

and

$$\mathbf{C}_x = \tilde{\mathbf{X}} \tilde{\mathbf{X}}^H. \quad (7.6)$$

Substituting  $\mathbf{C}_x$  into (7.2), taking the logarithm and dropping constant terms, we have

$$L(\mathbf{S}_x^{-1}) = \ln \det [\mathbf{S}_x^{-1}] - \text{tr} [\mathbf{S}_x^{-1} \mathbf{C}_x], \quad (7.7)$$

so  $\mathbf{C}_x$  is a sufficient statistic to estimate  $\mathbf{S}_x^{-1}$ . Taking the matrix gradient of  $L(\mathbf{S}_x^{-1})$  (using (A.397) and (A.393)) and setting the result equal to zero gives

$$[\mathbf{S}_x^T - \mathbf{C}_x^T]_{\mathbf{S}_x = \hat{\mathbf{S}}_x} = \mathbf{0}. \quad (7.8)$$

If we do not impose any structure on  $\mathbf{S}_x$ , then

$$\hat{\mathbf{S}}_x = \mathbf{C}_x. \quad (7.9)$$

Thus, the maximum likelihood estimate of the spectral matrix is the sample correlation matrix. We observe that  $\mathbf{C}_x$  is Hermitian, and if  $K \geq N$ , it is positive definite.

The elements of the sample spectral matrix have a probability density given by the complex Wishart density (e.g., Goodman [Goo63], [Hay96], [And63], [And84], or [Mui82]).<sup>2</sup>

---

<sup>1</sup>We use both  $\mathbf{X}_k$  and  $\mathbf{X}(k)$  to denote the  $k$ th snapshot vector. The second version is more appropriate when we write the components of the vector.

<sup>2</sup>Our discussion is based on Kelly and Forsythe [KF89]. The appendices of [KF89] contain an extensive discussion of techniques for multivariate statistical analysis.

Defining

$$\mathbf{C}_K = K\mathbf{C}_{\mathbf{x}}, \quad (7.10)$$

one can show that

$$p_{\mathbf{C}_K}(\mathbf{C}_K) = \frac{|\mathbf{C}_K|^{K-N}}{\Gamma_N(K)|\mathbf{S}_{\mathbf{x}}|^K} etr \left[ -\mathbf{S}_{\mathbf{x}}^{-1}\mathbf{C}_K \right], \quad (7.11)$$

where

$$etr[\mathbf{A}] \triangleq \exp[tr\mathbf{A}], \quad (7.12)$$

and  $\Gamma_N(K)$  is a generalization of the Gamma function,

$$\Gamma_N(K) = \Pi^{N(N-1)/2} \prod_{j=0}^{N-1} \Gamma(K-j). \quad (7.13)$$

Note that  $\Gamma_1(K) = \Gamma(K)$ .

The probability density in (7.11) is referred to as the complex Wishart density and denoted by  $\mathcal{W}_N(K, \mathbf{S}_{\mathbf{x}})$  and is defined over a space of non-negative definite Hermitian matrices.

It is a generalization of the complex chi-squared density. If  $N$  equals one (a single element), then  $\mathbf{X}$  is a scalar,  $X$ , and

$$C_K = \sum_{k=1}^K |X(k)|^2, \quad (7.14)$$

and  $\mathbf{S}_{\mathbf{x}} = \sigma_X^{-2}$ . Defining

$$\chi^2(K) = \frac{C_K}{\sigma_X^{-2}}, \quad (7.15)$$

and using (7.11), the probability density of  $\chi^2(K)$  is

$$p_{\chi^2}(Y) = \frac{(Y)^{K-1}}{\Gamma(K)} \exp(-Y), \quad Y \geq 0, \quad (7.16)$$

which is familiar as the complex chi-squared probability density with  $K$  degrees of freedom. Several properties of the complex chi-squared probability density are derived in the problems.

The complex Wishart density has several properties that we use in the sequel (e.g., [KF89], [Hay96], [Mui82], [And84]).

We assume that  $\mathbf{C}_K$  has a complex Wishart density  $\mathcal{W}_N(K, \mathbf{S}_{\mathbf{x}})$ . Then:

1. Let  $\mathbf{a}$  be any  $N \times 1$  random vector that is independent of  $\mathbf{C}_K$  and the  $p(\mathbf{a}) = \mathbf{0} = 0$ . Then:

$$y_1 \triangleq \frac{\mathbf{a}^H \mathbf{C}_K \mathbf{a}}{\mathbf{a}^H \mathbf{S}_x \mathbf{a}}, \quad (7.17)$$

is chi-square distributed with  $K$  degrees of freedom and is independent of  $\mathbf{a}$ , and

$$y_2 \triangleq \frac{\mathbf{a}^H \mathbf{S}_x^{-1} \mathbf{a}}{\mathbf{a}^H \mathbf{C}_K^{-1} \mathbf{a}}, \quad (7.18)$$

is chi-square distributed with  $K - N + 1$  degrees of freedom.

2. Let  $\mathbf{B}$  be a  $N \times M$  matrix of rank  $M$ , then:

- (a)  $\mathbf{B}^H \mathbf{C}_K \mathbf{B}$  is  $\mathcal{W}_M(K, \mathbf{B}^H \mathbf{S}_x \mathbf{B})$ . (This could correspond to a beamspace processor.)
- (b)  $\left[ \mathbf{B}^H \mathbf{C}_K \mathbf{B} \right]^{-1}$  is  $\mathcal{W}_M\left(K - N + M, \left[ \mathbf{B}^H \mathbf{C}_K^{-1} \mathbf{B} \right]^{-1}\right)$ .

We can also show that, if  $\mathbf{b}$  is a fixed non-zero vector, then

$$\begin{aligned} E\left[\mathbf{b}^H \mathbf{C}_K^{-1} \mathbf{b}\right] &= \left(\mathbf{b}^H \mathbf{C}_K^{-1} \mathbf{b}\right) E\left[\frac{1}{\chi^2(K - N + 1)}\right] \\ &= \frac{\mathbf{b}^H \mathbf{C}_K^{-1} \mathbf{b}}{K - N - 1}, \quad K > N + 1. \end{aligned} \quad (7.19)$$

This result implies

$$E\left[\mathbf{C}_K^{-1}\right] = \frac{\mathbf{S}_x^{-1}}{K - N - 1}, \quad K > N + 1. \quad (7.20)$$

The statistical properties of  $\mathbf{C}_K$  as an estimator are discussed in various statistics texts (e.g., Anderson [And84]). We are normally interested in the statistical behavior of functions derived from  $\mathbf{C}_K$ , so we focus our attention on their behavior. In the next section, we consider the statistical behavior of the eigenvectors and eigenvalues.

### 7.2.2 Asymptotic Behavior

In many applications we do an eigendecomposition of the estimated spectral matrix  $\hat{\mathbf{S}}_x$ ,

$$\hat{\mathbf{S}}_x = \sum_{i=1}^N \hat{\lambda}_i \hat{\Phi}_i \hat{\Phi}_i^H. \quad (7.21)$$

We want to investigate the properties of the  $\hat{\lambda}_i$  and  $\hat{\Phi}_i$  for large  $K$ . The original results are due to Anderson [And63] and were extended to the complex case by Gupta [Gup65] (e.g., [Bri81], [Wil65], [KW91]).

One can show that the eigenvalues  $\hat{\lambda}_i$  are asymptotically Gaussian and independent of the eigenvectors  $\hat{\Phi}_i$  for  $i = 1, 2, \dots, N$ . In addition,

$$E[\hat{\lambda}_i] = \lambda_i + O(K^{-1}), \quad (7.22)$$

and

$$Cov[\hat{\lambda}_i, \hat{\lambda}_j] = \delta_{ij} \frac{\lambda_i^2}{K} + O(K^{-2}). \quad (7.23)$$

The following results concerning the estimated eigenvectors can be derived (e.g., [KW91]). Define

$$\hat{\Phi}_i = \Phi_i + \eta_i. \quad (7.24)$$

Then

$$E[\eta_i] = -\frac{\lambda_i}{2K} \sum_{\substack{k=1 \\ k \neq i}}^N \frac{\lambda_k}{(\lambda_i - \lambda_k)^2} \Phi_i \triangleq c_i \Phi_i, \quad (7.25)$$

$$E[\eta_i \eta_j^H] = \frac{\lambda_i}{K} \sum_{\substack{k=1 \\ k \neq i}}^N \frac{\lambda_k}{(\lambda_i - \lambda_k)^2} \Phi_k \Phi_k^H \delta_{ij}, \quad (7.26)$$

$$E[\eta_i \eta_j^T] = -\frac{\lambda_i \lambda_j}{K (\lambda_i - \lambda_j)^2} \Phi_j \Phi_i^T (1 - \delta_{ij}). \quad (7.27)$$

We use these results later in the text to analyze the asymptotic behavior of various processing algorithms.

### 7.2.3 Forward–Backward Averaging

The sample correlation matrix is widely used as an estimate of  $\mathbf{S}_x$ . When the array manifold vector is conjugate symmetric,

$$\mathbf{v}_\psi(\psi) = \mathbf{J} \mathbf{v}_\psi^*(\psi), \quad (7.28)$$

then we can introduce constraints in the estimator to improve performance. We discuss this technique in this section.

To illustrate the procedure, we use a linear array with an even number of elements placed symmetrically about the origin. We index the

elements from 1 to  $N$ . We do not require that they be uniformly spaced. Thus,

$$d_i = -d_{N+1-i}. \quad (7.29)$$

The  $k$ th snapshot from the array when the input is a plane wave from DOA  $\theta$  is,

$$\mathbf{X}(k) = f(k) \left[ e^{-j\frac{2\pi}{\lambda}d_1 \cos \theta} \mid \dots \mid e^{-j\frac{2\pi}{\lambda}d_N \cos \theta} \right]^T + \mathbf{w}(k), \quad (7.30)$$

where  $f(k)$  is a zero-mean complex Gaussian variable and  $\mathbf{w}(k)$  is a complex Gaussian noise vector.

The spectral matrix is

$$\mathbf{S}_x = E \left[ \mathbf{X}(k) \mathbf{X}^H(k) \right]. \quad (7.31)$$

The  $ij$  element is,

$$[\mathbf{S}_x]_{ij} = \sigma_s^2 \left[ e^{-j\frac{2\pi}{\lambda}(d_i - d_j) \cos \theta} \right] + \sigma_w^2 \delta_{ij}. \quad (7.32)$$

Using (7.29) in (7.32) we have

$$[\mathbf{S}_x]_{N+1-i, N+1-j} = \left\{ [\mathbf{S}_x]_{ij} \right\}^*. \quad (7.33)$$

Thus,  $\mathbf{S}_x$  is a centrohermitian matrix (A.138). It is also a Hermitian matrix,

$$\mathbf{S}_x^H = \mathbf{S}_x. \quad (7.34)$$

Therefore, it is also persymmetric (A.134). As an example, for  $N = 4$ ,

$$\mathbf{S}_x = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{12}^* & s_{22} & s_{23} & s_{13} \\ s_{13}^* & s_{23}^* & s_{22} & s_{12} \\ s_{14}^* & s_{13}^* & s_{12}^* & s_{11} \end{bmatrix}. \quad (7.35)$$

The centrohermitian-Hermitian property implies that

$$\mathbf{S}_x = \mathbf{J} \mathbf{S}_x^* \mathbf{J}, \quad (7.36)$$

Where  $\mathbf{J}$  is the exchange matrix (e.g., (A.125) and (A.141)). Similarly,

$$\mathbf{S}_x^{-1} = \mathbf{J} \mathbf{S}_x^{*-1} \mathbf{J}. \quad (7.37)$$

The left  $\mathbf{J}$  reverses the rows and the right  $\mathbf{J}$  reverses the columns.

We want to find the maximum likelihood estimate of  $\mathbf{S}_x$  subject to the centrohermitian-Hermitian constraint.

The sample spectral matrix  $\mathbf{C}_x$  is defined in (7.3). We refer to the construction of  $\mathbf{C}_x$  as forward averaging. We now define a technique called backward averaging. The technique is implemented by reversing and conjugating the snapshot vector  $\mathbf{X}[k]$ . Define

$$\mathbf{X}^J = \mathbf{J}\mathbf{X}^*. \quad (7.38)$$

We define a backward averaged sample spectral matrix as

$$\mathbf{C}_{x,b} \triangleq \frac{1}{K} \sum_{k=1}^K \mathbf{J}\mathbf{X}_k^*\mathbf{X}_k^T\mathbf{J} = \mathbf{J}\mathbf{C}_x^*\mathbf{J}, \quad (7.39)$$

and FB averaged sample spectral matrix as

$$\mathbf{C}_{x,fb} = \frac{1}{2K} \sum_{k=1}^K \left( \mathbf{X}_k\mathbf{X}_k^H + \mathbf{J}\mathbf{X}_k^*\mathbf{X}_k^T\mathbf{J} \right) = \frac{1}{2} (\mathbf{C}_x + \mathbf{J}\mathbf{C}_x^*\mathbf{J}). \quad (7.40)$$

The joint probability density function was given in (7.2). We repeat (7.7) as,

$$L(\mathbf{S}_x^{-1}) = \ln \det [\mathbf{S}_x^{-1}] - \text{tr} [\mathbf{S}_x^{-1}\mathbf{C}_x]. \quad (7.41)$$

We maximize  $L(\mathbf{S}_x^{-1})$  subject to the constraint,

$$\hat{\mathbf{S}}_x = \mathbf{J}\hat{\mathbf{S}}_x^*\mathbf{J}. \quad (7.42)$$

Observe that

$$\begin{aligned} \text{tr} [\mathbf{S}_x^{-1}\mathbf{C}_x] &= \text{tr} [[\mathbf{J}\mathbf{S}_x^{*-1}\mathbf{J}]\mathbf{C}_x] \\ &= \text{tr} [\mathbf{S}_x^{*-1}\mathbf{J}\mathbf{C}_x\mathbf{J}] \\ &= \text{tr} [\mathbf{S}_x^{-1}\mathbf{J}\mathbf{C}_x\mathbf{J}], \end{aligned} \quad (7.43)$$

where the last equality follows because the trace is real. Using (7.43), the ln likelihood function in (7.41) can be written as

$$\begin{aligned} L(\mathbf{S}_x^{-1}) &= \ln \det [\mathbf{S}_x^{-1}] - \text{tr} \left[ \frac{1}{2} \mathbf{S}_x^{-1} [\mathbf{C}_x + \mathbf{J}\mathbf{C}_x^*\mathbf{J}] \right] \\ &= \ln \det [\mathbf{S}_x^{-1}] - \text{tr} [\mathbf{S}_x^{-1}\mathbf{C}_{x,fb}]. \end{aligned} \quad (7.44)$$

This equation has the same form as (7.7). Thus, the unconstrained maximum of (7.44) is  $\mathbf{C}_{\mathbf{x},fb}$  which is a centrohermitian-Hermitian matrix. Therefore,

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \mathbf{C}_{\mathbf{x},fb}. \quad (7.45)$$

is the constrained maximum likelihood estimate.<sup>3</sup>

We can also write (7.45) using the data matrix in (7.5) as

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \frac{1}{2} \left( \tilde{\mathbf{X}} \tilde{\mathbf{X}}^H + \mathbf{J} \tilde{\mathbf{X}}^* \tilde{\mathbf{X}}^T \mathbf{J} \right). \quad (7.46)$$

The result in (7.46) can also be written as

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \tilde{\mathbf{X}}_{fb} \tilde{\mathbf{X}}_{fb}^H, \quad (7.47)$$

where

$$\tilde{\mathbf{X}}_{fb} = \frac{1}{\sqrt{2}} \left[ \begin{array}{c|c} \tilde{\mathbf{X}} & \mathbf{J} \tilde{\mathbf{X}}^* \end{array} \right] \quad (7.48)$$

is an  $N \times 2K$  data matrix.

Although a linear array was used as an example, it is important to note that an array manifold vector for a large class of arrays can be written so that it is conjugate symmetric. The symmetry requirement on the array geometry is that for every element located at  $\mathbf{p}_n$ , there must be an identical element at  $-\mathbf{p}_n$ . Arrays that satisfy this condition include:

- (i) Standard rectangular array;
- (ii) Standard hexagonal array ( $N = 7, 19, 37, 61, \dots$ );
- (iii) Uniform circular array ( $N$  even);
- (iv) Uniform cylindrical array ( $N, M$  even);
- (v) Concentric circles array ( $N_i$  even).

The exploitation of centrohermitian-persymmetric matrices for communication problems was introduced by Cantoni and Butler [CB76]. They were used by Nitzberg [Nit80] for adaptive arrays and by Evans et al. [EJS82] for DOA estimation using arrays. Nuttall [Nut76] utilizes FB averaging in the context of linear prediction techniques.

---

<sup>3</sup>An early derivation of the result is due to Nitzberg [Nit80] who references Rao [Rao46] and Cantoni and Butler [CB76]. Our derivation is due to Jansson and Stoica [JS99].

$\hat{\mathbf{S}}_{\mathbf{x},fb}$  has properties that will be important in our signal processing discussion. We discuss three of these properties in this section and develop others at appropriate points in the text.<sup>4</sup>

**Property 1:** Complex FB spectral matrices can be transformed into real spectral matrices.

One of the operations that we will use frequently is

$$\hat{\mathbf{S}}_{12} = \mathbf{B}_1^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{B}_2, \quad (7.49)$$

where  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are  $N \times M$  matrices. As an example,  $\hat{\mathbf{S}}_{12}$  is the spectral matrix at the output of a beamspace transformation with

$$\mathbf{B}_{bs}^H = \mathbf{B}_1^H = \mathbf{B}_2^H. \quad (7.50)$$

If  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are both column conjugate symmetric,

$$\mathbf{B}_1 = \mathbf{J}\mathbf{B}_1^*, \quad \mathbf{B}_2 = \mathbf{J}\mathbf{B}_2^*, \quad (7.51)$$

or both column conjugate asymmetric,

$$\mathbf{B}_1 = -\mathbf{J}\mathbf{B}_1^*, \quad \mathbf{B}_2 = -\mathbf{J}\mathbf{B}_2^*, \quad (7.52)$$

then  $\hat{\mathbf{S}}_{12}$  is real.

This result follows directly from the definitions,

$$\begin{aligned} \mathbf{S}_{12}^* &= \left( \mathbf{B}_1^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{B}_2 \right)^* = \mathbf{B}_1^T \hat{\mathbf{S}}_{\mathbf{x},fb}^* \mathbf{B}_2^* = \mathbf{B}_1^T \left( \mathbf{J} \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{J} \right) \mathbf{B}_2^* \\ &= (\mathbf{J}\mathbf{B}_1)^T \hat{\mathbf{S}}_{\mathbf{x},fb} (\mathbf{J}\mathbf{B}_2^*) = \mathbf{B}_1^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{B}_2 = \mathbf{S}_{12}. \end{aligned} \quad (7.53)$$

As  $\hat{\mathbf{S}}_{12}$  and  $\hat{\mathbf{S}}_{12}^*$  are equal,  $\hat{\mathbf{S}}_{12}$  is real.

**Property 1A:** If the beamspace matrix  $\mathbf{B}_{bs}$  has columns that are either conjugate symmetric or conjugate asymmetric, then

$$Re\{\mathbf{B}_{bs}^H \hat{\mathbf{S}}_{\mathbf{x}} \mathbf{B}_{bs}\} = \mathbf{B}_{bs}^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{B}_{bs}. \quad (7.54)$$

Therefore, we can map the data  $\tilde{\mathbf{X}}$  into beamspace and compute,

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \tilde{\mathbf{X}}_{bs} \tilde{\mathbf{X}}_{bs}^H, \quad (7.55)$$

---

<sup>4</sup>The first part of our discussion follows Linebarger et al. [LDD94] with somewhat different notation.

and take the real part,

$$\hat{\mathbf{S}}_{\mathbf{x},bs,fb} = \text{Re}\{\hat{\mathbf{S}}_{\mathbf{x},bs}\}, \quad (7.56)$$

to obtain FB averaging in beamspace. The result in (7.54) follows by substituting the various matrices into (7.54) and using the conjugate symmetry properties.

The implication of this is that if we process the data using FB averaging and then use beamspace processing (Section 3.10), the resulting spectral matrix in beamspace will be real. Alternatively, we can process the data in beamspace without FB averaging and then take the real part of the estimated spectral matrix. The advantage of this result is that we can use real arithmetic for subsequent operations, which gives a computational saving of up to 75%. Computational advantages for various operations are discussed in detail in [LDD94] (see also [ZKS93]).

A particular beamspace processor of interest is the eigenvector beamspace processor in Figure 5.19. If  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  is a centrohermitian-persymmetric matrix (e.g., a linear array), then  $\frac{N}{2}$  eigenvectors will be conjugate symmetric and  $\frac{N}{2}$  eigenvectors will be conjugate asymmetric (see Section A.4.2.2 in Appendix A). Therefore, the beamspace spectral matrices will be real.

We also observe that the elements of  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  will not have a complex Wishart probability density. In [PK89], the probability density and various asymptotic properties are derived.

In many cases, we process the data in element space. Here we use a unitary transformation to obtain the advantage of real processing. The next two properties develop this technique.

**Property 2: Eigenvalues and eigenvectors:** Consider the product,

$$\mathbf{S}_{\mathbf{Q}} = \mathbf{Q}^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{Q}, \quad (7.57)$$

where  $\mathbf{Q}$  is unitary and column conjugate symmetric. If  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  has even dimension, then

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{J} & -j\mathbf{J} \end{bmatrix}, \quad (7.58)$$

where  $\mathbf{I}$  and  $\mathbf{J}$  have dimension  $N/2$ , is computationally attractive because of its sparse structure. If  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  has odd dimension,

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{0} & j\mathbf{I} \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ \mathbf{J} & \mathbf{0} & -j\mathbf{J} \end{bmatrix} \quad (7.59)$$

is used. Because  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  is centrohermitian and Hermitian,  $\mathbf{S}_Q$  is symmetric.

We consider the even-dimension case in the text. The eigenvector decomposition of  $\mathbf{S}_Q$  can be written as,

$$\mathbf{S}_Q = \mathbf{U}_c \Lambda_c \mathbf{U}_c^T, \quad (7.60)$$

where  $\mathbf{U}_c$  is an  $N \times N$  matrix of the eigenvectors

$$\mathbf{U}_c = \begin{bmatrix} \phi_{c1} & \phi_{c2} & \cdots & \phi_{cN} \end{bmatrix}, \quad (7.61)$$

and  $\Lambda_c$  is the diagonal matrix of eigenvalues,

$$\Lambda_c \triangleq \text{diag}\{\lambda_{c1}, \lambda_{c2}, \dots, \lambda_{cN}\}. \quad (7.62)$$

We can partition  $\mathbf{U}_c$  into an upper and lower half:

$$\mathbf{U}_c = \begin{bmatrix} \mathbf{U}_{c1} \\ \mathbf{U}_{c2} \end{bmatrix}. \quad (7.63)$$

Then (7.60) can be written as

$$\mathbf{S}_Q = \begin{bmatrix} \mathbf{U}_{c1} \\ \mathbf{U}_{c2} \end{bmatrix} \Lambda_c \begin{bmatrix} \mathbf{U}_{c1}^T & \mathbf{U}_{c2}^T \end{bmatrix}. \quad (7.64)$$

Since  $\mathbf{Q}$  is unitary, the eigenvalues of  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  will be the same as the eigenvalues of  $\mathbf{S}_Q$ . Thus, we can write the eigendecomposition of  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  as

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \mathbf{U}_R \Lambda_c \mathbf{U}_R^T. \quad (7.65)$$

The eigenvectors of  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  follow directly from the unitary property of  $\mathbf{Q}$ . The relation (7.57) can be written as

$$\mathbf{Q} \mathbf{S}_Q \mathbf{Q}^H = \mathbf{Q} \mathbf{Q}^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{Q} \mathbf{Q}^H = \hat{\mathbf{S}}_{\mathbf{x},fb}. \quad (7.66)$$

Thus,

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \mathbf{Q} \mathbf{U}_c \Lambda_c \mathbf{U}_c^T \mathbf{Q}^H, \quad (7.67)$$

so

$$\mathbf{U}_R = \mathbf{Q} \mathbf{U}_c = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{U}_{c1} + j\mathbf{U}_{c2} \\ \mathbf{J}(\mathbf{U}_{c1} - j\mathbf{U}_{c2}) \end{bmatrix}. \quad (7.68)$$

Note that the transformation  $\mathbf{Q}$  has a block identity-exchange matrix structure. Therefore, both the transformation in (7.57) and the inverse

transformation (7.65) involve only additions rather than general matrix multiplication.

**Property 3: Efficient construction of  $\hat{\mathbf{S}}_{\mathbf{x},fb}$ :** The FB sample correlation matrix is given by (7.46) as

$$\hat{\mathbf{S}}_{\mathbf{x},fb} = \frac{1}{2} (\tilde{\mathbf{X}} \tilde{\mathbf{X}}^H + \mathbf{J} \tilde{\mathbf{X}}^* \tilde{\mathbf{X}}^T \mathbf{J}) = \tilde{\mathbf{X}}_{fb} \tilde{\mathbf{X}}_{fb}^H, \quad (7.69)$$

where  $\tilde{\mathbf{X}}_{fb}$  is defined in (7.48). We now write,

$$\mathbf{S}_{\mathbf{Q}} = \mathbf{Q}^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{Q} = \mathbf{Q}^H \tilde{\mathbf{X}}_{fb} \tilde{\mathbf{X}}_{fb}^H \mathbf{Q} \triangleq \mathbf{Z}_{fb} \mathbf{Z}_{fb}^T, \quad (7.70)$$

where

$$\mathbf{Z}_{fb} = \mathbf{Q}^H \tilde{\mathbf{X}}_{fb} \mathbf{L}, \quad (7.71)$$

and  $\mathbf{L}$  is a unitary transform defined as

$$\mathbf{L} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix}, \quad (7.72)$$

and  $\mathbf{Z}_{fb}$  is real.

For  $N$  even, we partition  $\tilde{\mathbf{X}}$  into two  $(N/2) \times K$  matrices,

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{X}}_1 \\ \tilde{\mathbf{X}}_2 \end{bmatrix}. \quad (7.73)$$

Using (7.73) along with (7.69)–(7.72), we obtain

$$\mathbf{Z}_{fb} = \begin{bmatrix} Re(\tilde{\mathbf{X}}_1 + \mathbf{J}\tilde{\mathbf{X}}_2) & -Im(\tilde{\mathbf{X}}_1 + \mathbf{J}\tilde{\mathbf{X}}_2) \\ Im(\tilde{\mathbf{X}}_1 - \mathbf{J}\tilde{\mathbf{X}}_2) & Re(\tilde{\mathbf{X}}_1 - \mathbf{J}\tilde{\mathbf{X}}_2) \end{bmatrix}, \quad (7.74)$$

which is real.

The steps to construct  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  can be summarized:

1. Construct  $\mathbf{Z}_{fb}$  using the forward-only data matrix in (7.74) [2NK real additions].
2. Construct  $\mathbf{S}_{\mathbf{Q}}$  using (7.70) [ $2N^2K$  real multiplications and  $2N^2K$  real additions].
3. Construct  $\hat{\mathbf{S}}_{\mathbf{x},fb}$  using (7.66) [ $N^2$  real additions].

Note that, in this case, we use the unitary transformations to get a real data matrix before performing any operations. We will find these techniques to be useful in many applications.

### 7.2.4 Structured Spectral Matrix Estimation

The derivation in the previous section exploited the centrosymmetric per-symmetric structure of  $\mathbf{S}_x$  in order to find a maximum likelihood estimate. For a uniform linear array,  $\mathbf{S}_x$  is also Toeplitz. We would like to exploit that structure to find a maximum likelihood estimate of  $\mathbf{S}_x$ .

This problem was introduced by Burg et al. [BLW82] and has been studied extensively in the literature (e.g., [And73], [Deg87], [Cad88], [WH88], [VWV88], [FM88], [Fuh88], [FTM88], [DMS89], [MFOS91], [Fuh91], [WJ93], [TM94], [FM97], [LSL98]). Li et al. [LSL99] derive a computationally efficient technique that provides an asymptotic ML estimate of a structured spectral matrix. In most of our applications, we use either  $\hat{\mathbf{S}}_x$  or  $\hat{\mathbf{S}}_{x,fb}$  as the estimate, so the reader is referred to this literature for a discussion of the issues.

### 7.2.5 Parametric Spatial Spectral Matrix Estimation

In many applications, we can construct a parametric model of  $\mathbf{S}_x$  and then estimate the parameters in order to construct the estimate  $\hat{\mathbf{S}}_x$ . We consider two parametric models that are widely used.

In the first model, we assume the input consists of  $D$  uncorrelated plane waves plus additive white noise. Then,

$$\mathbf{S}_x = \mathbf{V}(\psi) \mathbf{S}_f \mathbf{V}^H(\psi) + \sigma_w^2 \mathbf{I}, \quad (7.75)$$

where  $\mathbf{V}(\psi)$  is an  $N \times D$  matrix composed of the array manifold vectors,

$$\mathbf{V}(\psi) = \begin{bmatrix} \mathbf{v}(\psi_1) & \mathbf{v}(\psi_2) & \cdots & \mathbf{v}(\psi_D) \end{bmatrix}, \quad (7.76)$$

and  $\mathbf{S}_f$  is a diagonal matrix of signal powers,

$$\mathbf{S}_f = \text{diag} \left[ \sigma_1^2, \sigma_2^2, \dots, \sigma_D^2 \right]. \quad (7.77)$$

We find a maximum likelihood estimate of the  $2D + 1$  parameters,

$$\theta \triangleq \{ \psi_1, \psi_2, \dots, \psi_D, \sigma_1^2, \sigma_2^2, \dots, \sigma_D^2, \sigma_w^2 \}. \quad (7.78)$$

We use these estimates in (7.75) to obtain

$$\hat{\mathbf{S}}_x = \mathbf{V}(\hat{\psi}) \hat{\mathbf{S}}_f \mathbf{V}^H(\hat{\psi}) + \hat{\sigma}_w^2 \mathbf{I}. \quad (7.79)$$

We discuss this technique in Chapter 8 after we have studied parameter estimation.

The second type of parametric model is the parametric wavenumber model introduced in Section 5.6. It is applicable to standard linear arrays. Here we estimate the parameters in the model and use these estimates to construct  $\hat{\mathbf{S}}_{\mathbf{x}}$ . This problem is the dual of the temporal spectral estimation problem, and the techniques are discussed extensively in the literature (e.g., books by Kay [Kay88], Marple [Mar87], and Stoica and Moses [SM97]). We do not develop this general approach in the text. A few simple cases are covered in the problem section of Chapter 8.

### 7.2.6 Singular Value Decomposition

A significant portion of our subsequent development will deal with the spatial spectral matrix of the received waveform at the sensors. In order to estimate the spatial spectral matrix, we form the sample spectral matrix

$$\hat{\mathbf{S}}_{\mathbf{x}} = \mathbf{C}_{\mathbf{x}} = \frac{1}{K} \sum_{k=1}^K \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.80)$$

If the array manifold vector is conjugate symmetric, we can use the FB averaging technique in (7.46)–(7.48) to construct the sample spectral matrix.

In many applications, we want to find the eigenvalues and eigenvectors of  $\hat{\mathbf{S}}_{\mathbf{x}}$  or  $\hat{\mathbf{S}}_{\mathbf{x},fb}$ . The SVD technique enables us to find the eigenvalues and eigenvectors directly from the data matrix  $\tilde{\mathbf{X}}$  in (7.4) or  $\tilde{\mathbf{X}}_{fb}$  in (7.48). The technique is developed in Section A.5.

There are important computational reasons for working directly with the data matrix rather than the sample spectral matrix. The dynamic range required to deal with  $\hat{\mathbf{S}}_{\mathbf{x}}$  is doubled. Thus, for a specified numerical accuracy, the required word length is doubled.

There are several efficient computational schemes for computing the SVD. All of the various computational programs such as LINPACK, EISPACK, and MATLAB have SVD algorithms included. SVD is widely used in a number of signal processing applications, and there is extensive literature on the topic (e.g., Deprettere [Dep88], in particular the tutorials in Part I; the discussion in Chapter 11 of Haykin [Hay96] and several of his references; Klema and Laub [KL80]; and Eckhart and Young [EY36]).

### 7.2.7 Summary

In this section, we have developed techniques for estimating the spectral matrix of the snapshot vector at the sensor output.

In Section 7.2.1, we introduced the (forward-only) sample spectral matrix and showed that, if we did not impose any structure on the spectral matrix, then  $\mathbf{C}_x$  is the maximum likelihood estimate of  $\mathbf{S}_x$ . The statistics of the sample spectral matrix are described by the complex Wishart probability density in (7.11).

In many applications, we use the eigenvectors and/or eigenvalues of the sample spectral matrix. Even when we do not use the eigenvectors and eigenvalues directly, they are often the key to the performance. In Section 7.2.2, expressions for the asymptotic behavior of the eigenvectors and eigenvalues were given. Later these results will be used to evaluate the performance of various array processing algorithms.

In Section 7.2.3, the property that many of the arrays that are used in practice have array manifold vectors that are conjugate symmetric was exploited. This property leads to a centrosymmetric Hermitian spectral matrix. In this case the maximum likelihood estimate is given by a FB averaged sample spectral matrix  $\mathbf{C}_{x,fb}$ . This property not only provides performance improvement but allows the use of real computation.

In Section 7.2.4, the idea of structured spectral estimation was introduced but not developed.

In Section 7.2.5, the idea of parametric spatial spectral matrix estimation was introduced but not developed.

In Section 7.2.6, we discussed the SVD technique that is developed in detail in Section A.5. This technique allowed us to find the eigenvalues and eigenvectors of the sample covariance matrix directly from the data.

This section provides the foundation for the adaptive beamforming algorithms that are developed in the remainder of the chapter. In some algorithms, we will utilize  $\hat{\mathbf{S}}_x$  or the eigenvalues and eigenvectors explicitly. In other algorithms, they are used implicitly. However, they always have a major influence on the performance of the algorithm.

### 7.3 Sample Matrix Inversion (SMI)

In adaptive beamformers, the sample spectral matrix  $\hat{\mathbf{S}}_x$  is estimated from the data samples. We recall from Section 7.2, that the maximum likelihood estimate for an unstructured matrix is just the sample spectral matrix. From (7.9),

$$\hat{\mathbf{S}}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.81)$$

In some applications, we can observe the input without the signal being present. For example, in the interval between returned pulses in an active radar. Then, we can construct an estimate of  $\mathbf{S}_n$  in the same manner,

$$\hat{\mathbf{S}}_n = \frac{1}{K} \sum_{k=1}^K \mathbf{N}(k) \mathbf{N}^H(k). \quad (7.82)$$

If  $\hat{\mathbf{S}}_n$  is available, we use it in place of  $\mathbf{S}_n$  in the MVDR beamformer (see (6.14) and (6.15)) to obtain

$$\boxed{\mathbf{w}_{mvdr,smi}^H \triangleq \Lambda_{smi} \mathbf{v}_s^H \hat{\mathbf{S}}_n^{-1}}, \quad (7.83)$$

and

$$\Lambda_{smi} = \left( \mathbf{v}_s^H \hat{\mathbf{S}}_n^{-1} \mathbf{v}_s \right)^{-1}. \quad (7.84)$$

The presence of  $\hat{\mathbf{S}}_n^{-1}$  gives rise to the name **sample matrix inversion** (SMI) for this approach.

If only  $\hat{\mathbf{S}}_x^{-1}$  is available, we use (7.81) in (6.71) to obtain

$$\boxed{\mathbf{w}_{mpdr,smi}^H \triangleq \frac{\mathbf{v}_m^H \hat{\mathbf{S}}_x^{-1}}{\mathbf{v}_m^H \hat{\mathbf{S}}_x^{-1} \mathbf{v}_m}}, \quad (7.85)$$

where  $\mathbf{v}_m$  is the nominal signal direction.

If the array is conjugate symmetric, we can use  $\hat{\mathbf{S}}_{x,fb}$  as an estimate of  $\mathbf{S}_x$  (or  $\mathbf{S}_n$ ).

In Figure 7.1, we show a diagram of the SMI beamformer. In this configuration, we are processing the data in blocks of size  $K$ . In Figure 7.2, we show a diagram of the SMI beamformer in the generalized sidelobe canceller configuration.

The equations specifying the SMI GSC implementation are

$$\hat{\mathbf{S}}_z(K) = \frac{1}{K} \sum_{k=1}^K \mathbf{Z}(k) \mathbf{Z}^H(k) = \mathbf{B}^H \hat{\mathbf{S}}_x(K) \mathbf{B}, \quad (7.86)$$

$$\hat{\mathbf{S}}_{zy_c^*}(K) = \frac{1}{K} \sum_{k=1}^K \mathbf{Z}(k) Y_c^*(k) = \mathbf{B}^H \hat{\mathbf{S}}_x(K) \mathbf{w}_q, \quad (7.87)$$

and

$$\begin{aligned} \hat{\mathbf{w}}_a(K) &= \hat{\mathbf{S}}_z^{-1}(K) \hat{\mathbf{S}}_{zy_c^*}(K) \\ &= [\mathbf{B}^H \hat{\mathbf{S}}_x(K) \mathbf{B}]^{-1} \mathbf{B}^H \hat{\mathbf{S}}_x(K) \mathbf{w}_q. \end{aligned} \quad (7.88)$$

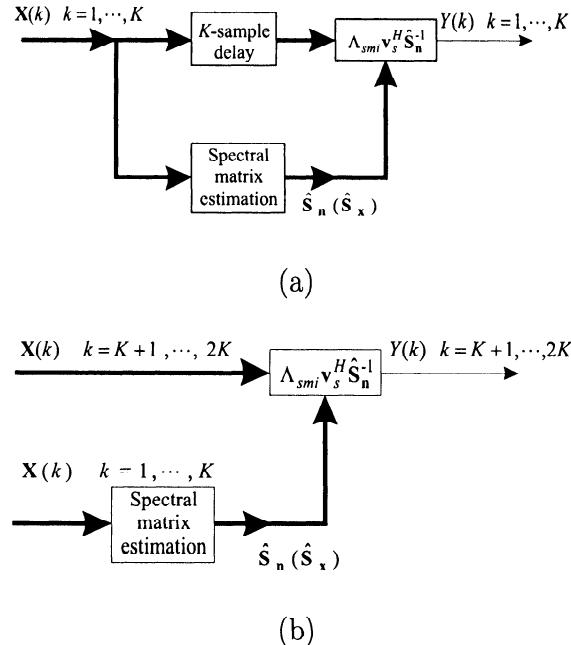


Figure 7.1 Diagram of the SMI beamformer: (a) use of same data vector; (b) use of new data vector.

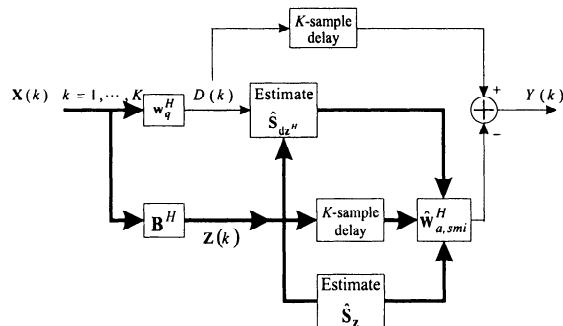


Figure 7.2 Diagram of the SMI beamformer in GSC configuration.

In Section 7.3.1, we discuss the statistical behavior of the  $SINR_{smi}$  as a function of the number of samples  $K$ . This behavior will indicate how quickly the performance of MVDR beamformer converges to the results in Chapter 6, which assumed perfect knowledge of either  $\mathbf{S}_x$  or  $\mathbf{S}_n$ .

In Section 7.3.2, we discuss how the SMI MVDR beamformer can be implemented in a recursive manner.

In Section 7.3.3, we introduce the technique of diagonal loading in which we add a constant diagonal matrix to  $\hat{\mathbf{S}}_x$  or  $\hat{\mathbf{S}}_n$  before utilizing them in the weight vector computation. This technique is called **regularization** in the statistical literature.

In Section 7.3.4, we revisit the case of conjugate-symmetric array manifold vectors and show how the implementation can be simplified when we use FB averaging.

### 7.3.1 $SINR_{smi}$ Behavior: MVDR and MPDR

In this section we discuss the  $SINR$  behavior of MVDR and MPDR beamformers as a function of  $K$ .

We first consider the beamformer in (7.83). We denote the weight vector using  $K$  samples as

$$\hat{\mathbf{w}}_{mvdr,smi}^H(K) \triangleq \Lambda_{smi}(K) \mathbf{v}_s^H \hat{\mathbf{S}}_n^{-1}(K). \quad (7.89)$$

When the context is clear, we suppress the subscript and use  $\hat{\mathbf{w}}(K)$ . We want to investigate the  $SINR$  behavior as a function of  $K$ . We assume that  $\mathbf{v}_s$  is known. The signal, interference, and noise inputs are sample functions of discrete time random processes so the  $SINR_o$  is a random variable. The  $SINR_o$  on the  $l$ th trial at sample  $K$  is

$$SINR_o(l) = \frac{\hat{\mathbf{w}}^H(K) \mathbf{v}_s f_l(K) f_l^H(K) \mathbf{v}_s^H \hat{\mathbf{w}}(K)}{\hat{\mathbf{w}}^H(K) [\mathbf{n}_l(K) \mathbf{n}_l^H(K)] \hat{\mathbf{w}}(K)}, \quad (7.90)$$

where  $f_l(K)$  and  $\mathbf{n}_l(K)$  are the sample values on the  $l$ th trial. The vector  $\mathbf{n}_l(K)$  contains the interference and the white sensor noise.

In Chapter 6, the “noise” was assumed to contain both the white sensor noise and the interference. Even though the noise vector  $\mathbf{n}$  contains both the white sensor noise and any additional interference, we use  $SINR_o$  in several sections in Chapter 7, to be consistent with the adaptive beamformer literature.

We also define a  $SINR_{smi}$  in which the  $\mathbf{n}_l(K)\mathbf{n}_l^H(K)$  term in (7.90) is replaced by  $\mathbf{S}_n$ ,

$$SINR_{smi} \triangleq \frac{\sigma_s^2 |\hat{\mathbf{w}}^H(K)\mathbf{v}_s|^2}{\hat{\mathbf{w}}^H(K)\mathbf{S}_n\hat{\mathbf{w}}(K)}. \quad (7.91)$$

The expression in (7.91) has a smoother behavior and is the definition that is normally used in the literature. We define a random variable  $\rho(K)$ ,

$$\rho(K) \triangleq \frac{SINR_{smi}(K)}{SINR_{mvdr}}, \quad 0 \leq \rho(K) \leq 1, \quad (7.92)$$

which is the ratio of the  $SINR$  due to the use of  $\hat{\mathbf{S}}_n$  to the  $SINR$  using  $\mathbf{S}_n$ . For simplicity, we suppress the  $K$  dependence and write

$$\rho \triangleq \frac{SINR_{smi}}{SINR_{mvdr}}, \quad 0 \leq \rho \leq 1. \quad (7.93)$$

We want to find the probability density of  $\rho$  and its mean and variance. The solution is given by Reed et al. [RMB74]. They indicate that their development is based in part on Capon and Goodman [CG70], Goodman [Goo63], and unpublished work by Goodman.

The development is lengthy, so we quote the result. Using the properties of the complex Wishart density (see Section 7.2.1), they show that  $\rho$  has a beta probability density,

$$p(\rho) = \frac{K!}{(N-2)!(K+1-N)!} (1-\rho)^{N-2} \rho^{K+1-N}, \quad 0 \leq \rho \leq 1. \quad (7.94)$$

This probability density is valid for  $N \geq 2$  and  $K \geq N$ . In (7.94),  $N$  is the number of elements. The probability density is shown in Figure 7.3 for various values of  $K$ .

We observe that the probability density does not depend on the signal strength or the noise and interference environment.

The mean of  $\rho$  is

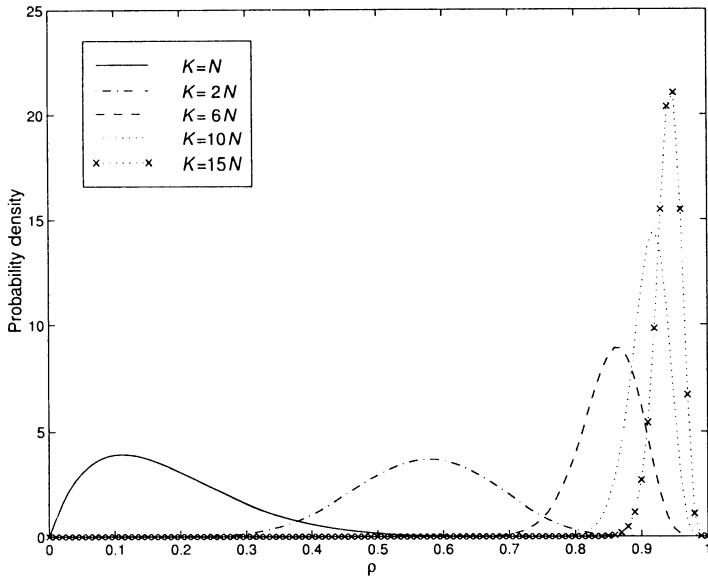
$$E[\rho] = \frac{K+2-N}{K+1}, \quad (7.95)$$

and the variance is

$$Var[\rho] = \frac{(K+2-N)(N-1)}{(K+1)^2(K+2)}. \quad (7.96)$$

Therefore,

$$E[SINR_{smi}] = \frac{K+2-N}{K+1} SINR_{mvdr}. \quad (7.97)$$

Figure 7.3 Probability density of  $\rho$ .

If we desire that  $E[\text{SINR}_{\text{smt}}] = \alpha \text{SINR}_{\text{mvdr}}$ , then we require,

$$K = \frac{1}{1-\alpha}(N - 2 + \alpha) \simeq \frac{1}{1-\alpha}N. \quad (7.98)$$

Thus,  $K = 2N - 3$  obtains an  $E[\text{SINR}_{\text{smt}}]$  that is 3 dB lower. To achieve  $\alpha = 0.95$  requires  $K = 20N$ . In practice, a common rule of thumb is that  $K = 2N$  samples are required for “satisfactory” performance.

When  $K = 2N - 3$ ,  $p(\rho)$  is symmetric about 0.5, so  $Pr[\rho < 0.5]$  equals 0.5. Boronson [Bor80] suggests that a useful measure is to determine the value of  $K$  such that

$$P[\rho < 1 - \delta] < \epsilon. \quad (7.99)$$

One can show [AS65] that

$$P[\rho < 1 - \delta] = \sum_{m=0}^{N-2} b(m; K, \delta), \quad (7.100)$$

where

$$b(m; K, \delta) = \binom{K}{m} \delta^m (1 - \delta)^{K-m}. \quad (7.101)$$

Then, assuming  $N > 3$ ,  $K \geq 3N$  implies

$$P(\rho < 0.5) < 0.0196, \quad (7.102)$$

and  $K \geq 4N$  implies

$$P(\rho < 0.5) < 0.0032. \quad (7.103)$$

We simulate the behavior of the MVDR beamformer using a Monte Carlo procedure. We denote the average of  $\rho$  over the trials by  $\bar{\rho}$ ,

$$\bar{\rho} \triangleq \frac{\overline{SINR}_{smi}}{\overline{SINR}_{mvdr}}, \quad (7.104)$$

where

$$\overline{SINR}_{smi} = \frac{1}{L} \sum_{l=1}^L SINR_{smi}(l), \quad (7.105)$$

and  $SINR_{smi}(l)$  is the value of (7.91) on the  $l$ th-trial.

We illustrate the behavior with a simple example.

### Example 7.3.1

Consider a standard 10-element linear array. The signal arrives from  $u_s = 0$ , and there is a single interferer at  $u_I = 0.15$  with an  $INR = 10$  dB. In Figure 7.4, we plot  $\bar{\rho}$  for 200 trials for both  $\hat{\mathbf{S}}_n = \mathbf{C}_x$  and  $\hat{\mathbf{S}}_n = \mathbf{C}_{x,fb}$ . For the  $\hat{\mathbf{S}}_n = \mathbf{C}_x$  case we show  $E[\rho]$  as given by (7.95). For modest  $K$ , the use of  $\hat{\mathbf{S}}_n = \mathbf{C}_{x,fb}$  halves the number of required samples to achieve a given value of  $\rho$ . Note that, although we specify a signal and interference environment, the result does not depend on it.

Monzingo and Miller [MM80] extended the result to include the case of the signal present, and the beamformer is given by (7.85). We define  $\eta$  to be the ratio of the  $SINRs$ ,

$$\begin{aligned} \eta &= \frac{SINR_{mpdr,smi}}{SINR_{mpdr}} \\ &= \frac{\rho'}{SINR_{mpdr}(1 - \rho') + 1}, \end{aligned} \quad (7.106)$$

where  $\rho'$  has the same probability density as  $\rho$ . The expectation of  $\eta$  can be written as an infinite series,

$$\begin{aligned} E[\eta] &= \frac{a}{a+b} \left\{ 1 + \sum_{i=1}^{\infty} (-SINR_{mpdr})^i \left( \frac{b}{a+b+1} \right) \right. \\ &\quad \left. \cdot \left( \frac{b+1}{a+b+2} \right) \cdots \left( \frac{i+b-1}{a+b+i} \right) \right\}, \end{aligned} \quad (7.107)$$

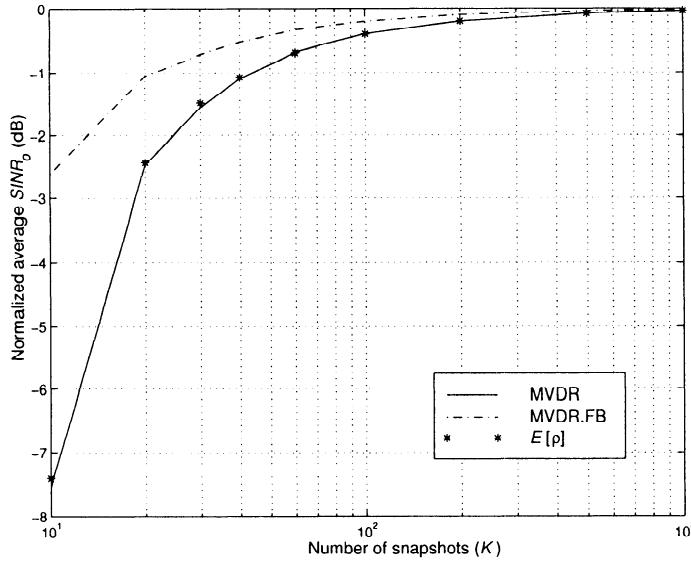


Figure 7.4 MVDR SMI beamformer:  $\bar{\rho}$  and  $E[\rho]$  versus  $K$ .

where

$$a = K - N + 2, \quad (7.108)$$

$$b = N - 1. \quad (7.109)$$

We can approximate the sum in the brace to obtain an approximate expression,<sup>5</sup>

$$E[\eta] \simeq \frac{a}{a+b} \left\{ \frac{1}{1 + SINR_{mpdr} \cdot \frac{b}{a+b+1}} \right\}. \quad (7.110)$$

We illustrate the results with the same signal and noise model as in Example 7.3.1.

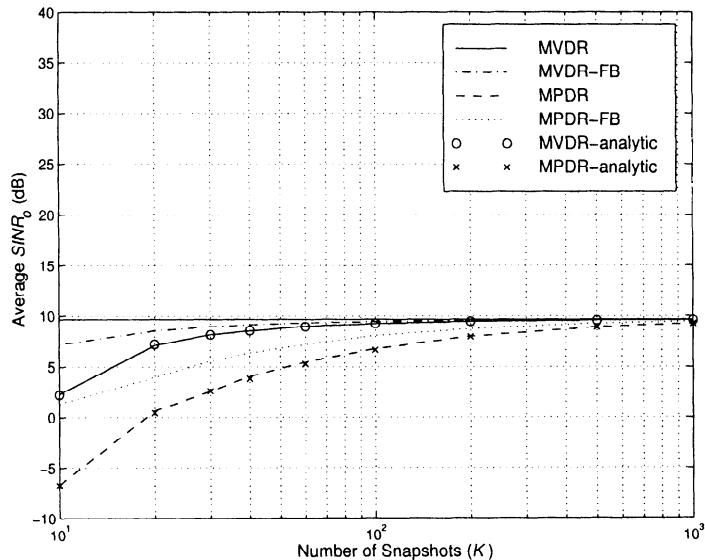
### Example 7.3.2

Consider a standard 10-element linear array. The signal arrives from  $u_s = 0$ . The SNR is varied from 0 dB to 30 dB in 10-dB steps. Two equal-power interferers arrive from  $u_I = 0.29$  and  $0.45$ , each with an INR = 20 dB.

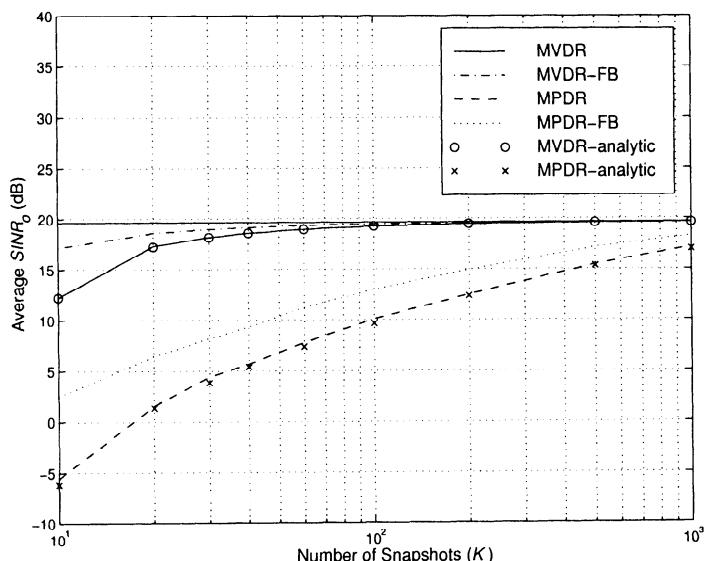
In Figure 7.5 we show the results of a Monte Carlo simulation. We plot the average  $SINR_o$  versus  $K$  for both  $\hat{\mathbf{S}}_x = \mathbf{C}_x$  and  $\hat{\mathbf{S}}_{x,fb} = \mathbf{C}_{x,fb}$ . We also show the analytic results using (7.97) and (7.110). We see that the presence of the signal in  $\hat{\mathbf{S}}_x$  causes a significant degradation in the performance of the SMI beamformer.

In Figure 7.6, we plot  $\bar{\eta}$ , which is the normalized quantity. This plot shows the effect of the signal level.

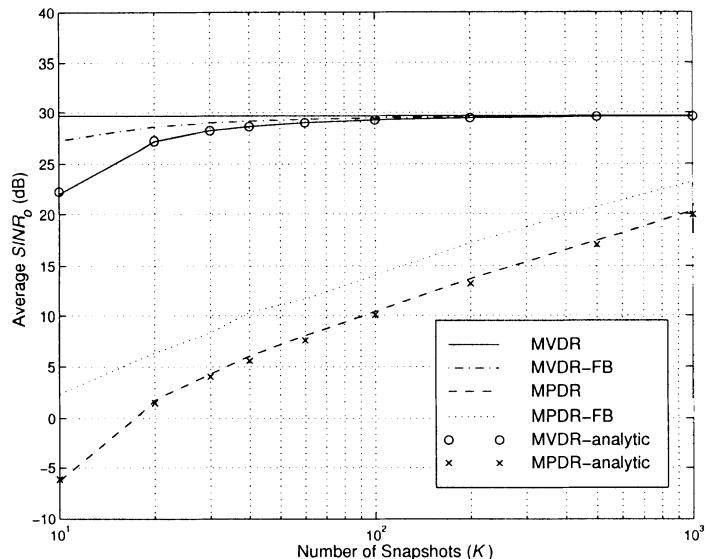
<sup>5</sup>Z. Tian, private communication.



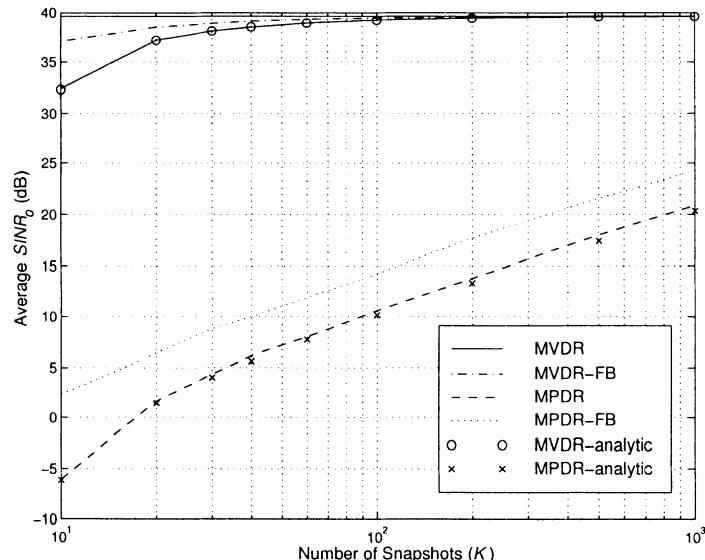
(a)



(b)



(c)



(d)

Figure 7.5 MVDR-SMI and MPDR-SMI beamformers: average  $SINR_o$  versus  $K$ : (a)  $SNR = 0$  dB; (b)  $SNR = 10$  dB; (c)  $SNR = 20$  dB; (d)  $SNR = 30$  dB.

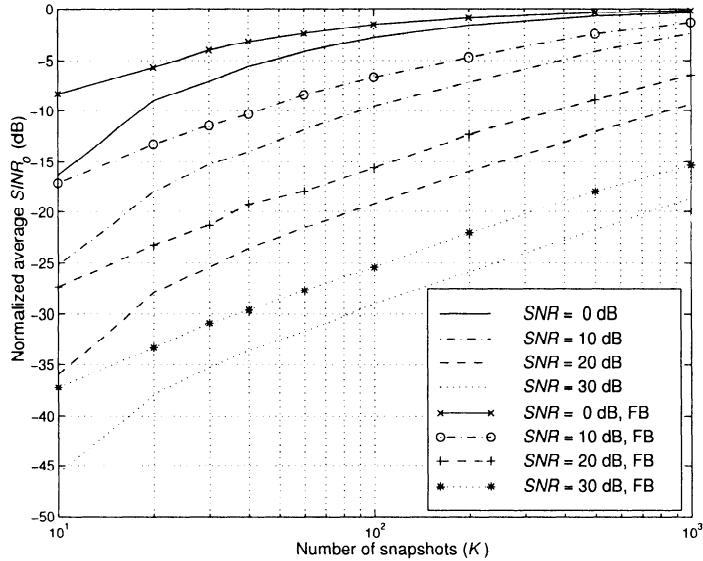


Figure 7.6 MPDR-SMI beamformer: normalized  $\overline{SINR}_{smi}(\bar{\eta})$  versus  $K$  for various  $SNR$ .

Feldman and Griffiths also develop an approximate expression for  $E[SINR_{mpdr,smi}]$  (see p.871 of [FG94]). The result is

$$E\{SINR_{mpdr}\} \approx \frac{SINR_{mpdr} \cdot K}{K + SINR_{mpdr} \cdot (N - 1)}. \quad (7.111)$$

This expression is valid when

$$N \gg 1,$$

$$K \gg N,$$

$$SINR_{mpdr} \gg N.$$

As  $K$  increases from  $N$  to  $\infty$ , the  $E[SINR_{mpdr,smi}]$  increases monotonically to  $SINR_{mpdr}$ .

From (7.111), we see that to be within 3 dB of the optimum value, we require

$$K = (SINR_{mpdr})(N - 1), \quad (7.112)$$

which will be significantly larger than the previous  $K = 2N$  result in most cases of interest.

### 7.3.2 LCMV and LCMP Beamformers

In the LCMV and LCMP beamformers developed in Section 6.7.2, we impose additional constraints on the beamformer and reduce the adaptive degrees of freedom. Monzingo and Miller [MM80] argue that we can use the results in (7.94) by defining  $(N - 1)$  as the number of adaptive degrees of freedom instead of the number of elements minus one. Van Veen ([VV91a], [VV91b], [HS92]) obtains a similar result (e.g., Reed et al. [RMB74], Capon and Goodman [CG70], Monzingo and Miller [MM80], Baggeroer [Bag76], Ganz et al. [GMW90], Kelly [Kel86], and Kelly and Forsythe [KF89]).

We consider a simple example to illustrate the behavior.

#### Example 7.3.3 (continuation, Example 7.3.2)

Consider a standard 10-element array. The signal-and-interference model are the same as in Example 7.3.2. We use an LCMP beamformer with derivative constraints (see Examples 6.7.2 and 6.7.10). The  $\mathbf{C}$  and  $\mathbf{g}^H$  matrices are given by (6.312) and (6.314), respectively:

$$\mathbf{C} = \begin{bmatrix} 1 & \vdots & \mathbf{d}(0) & \vdots & \ddot{\mathbf{d}}(0) \end{bmatrix}, \quad (7.113)$$

and

$$\mathbf{g}^H = \begin{bmatrix} 1 & \vdots & 0 & \vdots & \ddot{B}_c(0) \end{bmatrix}. \quad (7.114)$$

In Figure 7.7(a), we show  $\bar{\rho}$  obtained from a Monte Carlo simulation for an LCMV beamformer. We also show  $E[\rho]$  obtained from (7.95) with  $N = 8$ . Comparing these results to those in Figure 7.4, we see that, by reducing the available adaptive degrees of freedom, we have improved the normalized SMI performance. In Figure 7.7(b), we plot  $\bar{\eta}$ , the normalized average  $SINR_o$  versus  $K$  for an LCMP beamformer. Comparing these results to those in Figure 7.6, we see an improvement.

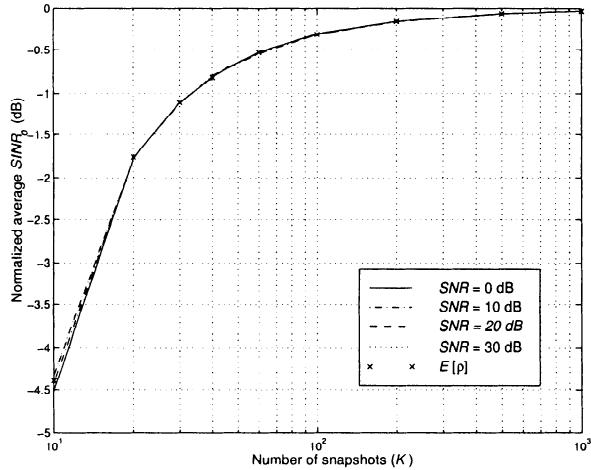
### 7.3.3 Fixed Diagonal Loading

The concept of diagonal loading is straightforward. We use,

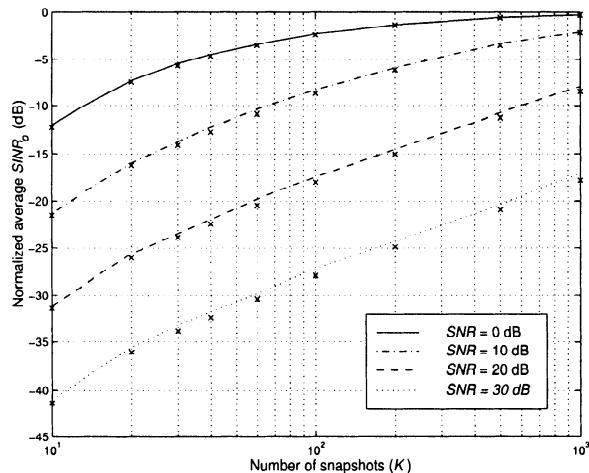
$$\tilde{\mathbf{S}}_{\mathbf{x},L} = \frac{1}{K} \sum_{k=1}^K \mathbf{X}(k)\mathbf{X}^H(k) + \sigma_L^2 \mathbf{I} \quad (7.115)$$

in place of the estimated spectral matrix in order to design  $\hat{\mathbf{w}}$ . We use  $\tilde{\mathbf{S}}_{\mathbf{x},L}$  instead of  $\hat{\mathbf{S}}_{\mathbf{x}}$  because it is not used as an estimate of  $\mathbf{S}_{\mathbf{x}}$ . We have added a diagonal matrix of level  $\sigma_L^2$ . We encountered diagonal loading in Sections 6.6.4 and 6.10 in the context of quadratically constrained beamformers. In this section, we use it for three purposes:

- (i) To improve the  $SINR_{smi}$  performance of the MPDR beamformer;
- (ii) To implement beamformers when  $K < N$ ;



(a)



(b)

Figure 7.7 LCMV-SMI and LCMP-SMI beamformers:  $u_s = 0$ ,  $u_I = 0.29$ , and 0.45,  $INR = 20 \text{ dB}$ ,  $SNR = 0, 10, 20, 30 \text{ dB}$ ; normalized  $\overline{SINR}_o$  versus  $K$ : (a) LCMV-SMI; (b) LCMP-SMI beamformers.

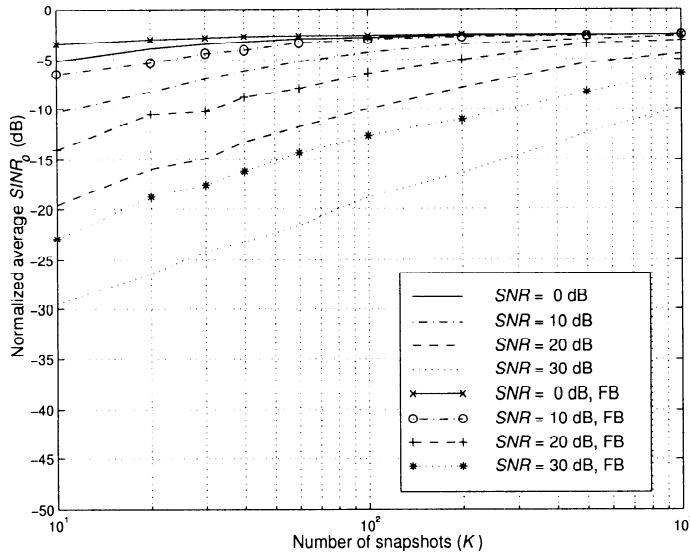


Figure 7.8 MPDR-SMI beamformer:  $INR = 20$  dB,  $\sigma_L^2/\sigma_W^2 = 20$  dB; normalized  $\overline{SINR}_o$  versus  $K$ .

- (iii) To achieve better sidelobe control and main-beam shaping in the SMI algorithm.

To demonstrate how diagonal loading improves the  $SINR_{smi}$  behavior, we consider the same model as in Section 7.3.1.

**Example 7.3.4** (continuation, Example 7.3.2)

Consider a standard 10-element linear array. The desired signal arrives from  $u_s = 0$  and two equal power interferers arrive from  $u_I = 0.29$  and  $0.45$ , each with an  $INR = 20$  dB. We use

$$\bar{\mathbf{S}}_{\mathbf{x},L} = \mathbf{C}_{\mathbf{x}} + \sigma_L^2 \mathbf{I}, \quad (7.116)$$

or

$$\bar{\mathbf{S}}_{\mathbf{x},L} = \mathbf{C}_{\mathbf{x},fb} + \sigma_L^2 \mathbf{I}. \quad (7.117)$$

in (7.85). In Figure 7.8, we plot the normalized average  $SINR_o$  versus  $K$  for  $\sigma_L^2 = 20$  dB. We see that there is significant improvement for all  $SNR$ . In Figure 7.9, we show the MVDR behavior. There is also improvement, but it is less significant.

When we use diagonal loading we can implement the beamformer for  $K < N$  because the  $\sigma_L^2 \mathbf{I}$  term makes  $\bar{\mathbf{S}}_{\mathbf{x},L}$  non-singular.

Several references (e.g., Hudson [Hud81], Gabriel [Gab85], and Brookner and Howell [BH85]) show that if there are  $D$  strong interferences and  $N \gg D$ , the beamformer can achieve effective nulling with  $K = 2D$  samples. With

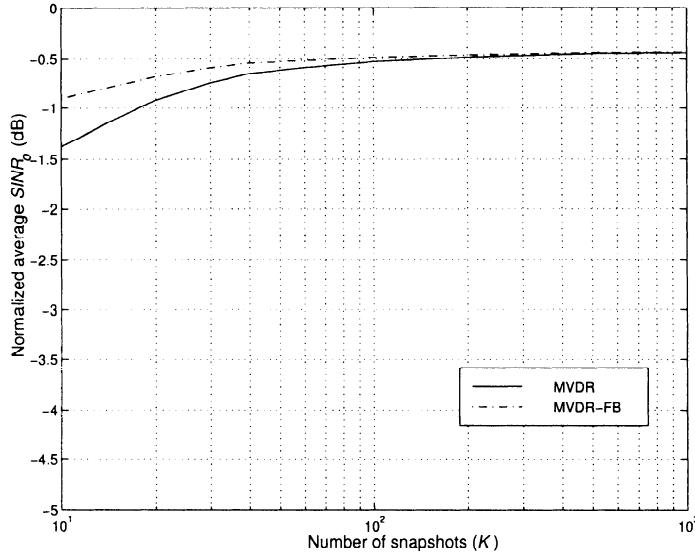


Figure 7.9 MVDR-SMI beamformer:  $INR = 20$  dB,  $\sigma_L^2/\sigma_W^2 = 20$  dB; normalized  $\overline{SINR}_o$  versus  $K$ .

FB averaging,  $K = D$ . We revisit Example 7.3.4 and explore the behavior for  $K = 2, 4$ , and  $6$ .

#### Example 7.3.4 (revisited)

Consider the same model as in Example 7.3.4. In Figure 7.10, we plot the normalized average  $SINR_o$  versus  $K$  for  $K \geq D$  for an MPDR-SMI beamformer with fixed diagonal loading.

We see that there is significant improvement in the  $\overline{SINR}_o$  due to diagonal loading with a small number of samples. However, it is also important to examine the beam pattern behavior. In Figure 7.11, we show the beam pattern for the two-interferer case ( $u_I = 0.29, 0.15$ ) with a high  $INR$  (20 dB), a modest  $SNR$  (10 dB), for  $K = 4$  using FB averaging. We vary  $\sigma_L^2/\sigma_w^2$  from  $-10$  dB to  $20$  dB in 10-dB steps. We observe that, although the nulls are deep and close to the interferer, a typical beam pattern has an undesirable sidelobe structure unless we use diagonal loading.

The same problem continues when  $K > N$ .

Kelly [Kel87a], [Kel87b] shows that the expected value of the sidelobes of the adapted pattern is

$$E[SLL] = \frac{1}{K+1}. \quad (7.118)$$

We consider a simple example to illustrate this effect.

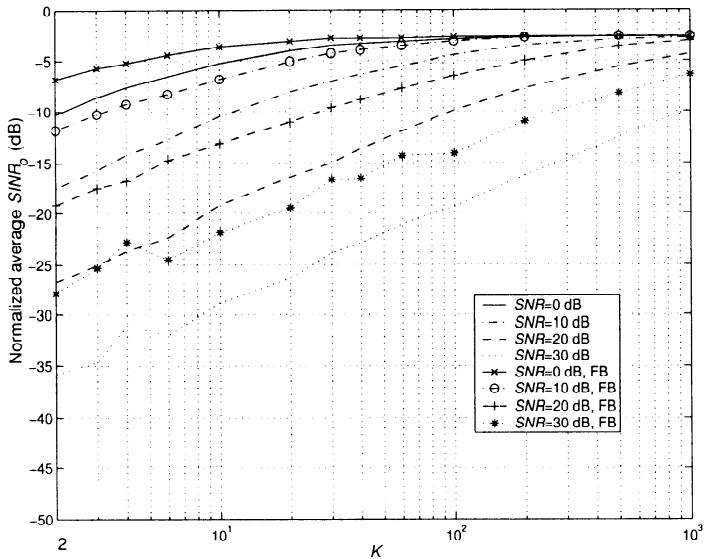


Figure 7.10 MPDR-SMI beamformer with fixed diagonal loading: normalized  $SINR_o$  versus  $K$ ,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB,  $LNR = \sigma_L^2/\sigma_W^2 = 20$  dB.

#### Example 7.3.5<sup>6</sup>

Consider a standard 10-element linear array. A Dolph-Chebychev weighting on the steering vector  $\mathbf{v}(\psi_s)$  provided a quiescent beam with -30-dB sidelobes. We use the constraint  $\mathbf{w}^H \mathbf{w}_q = 1$ . The beam is steered to broadside. A single plane-wave source with an  $INR$  of 30 dB impinges on the array at  $u_I = 0.29$ . There is no diagonal loading.

The results of a simulation for  $K = 2N$  and  $K = 6N$  are shown in Figure 7.12.

To understand the behavior we write the weighting vector using the eigenvector expansion of  $\hat{\mathbf{S}}_n$ .<sup>7</sup> We assume that  $\sigma_L^2 = 0$  and that there are  $D$  plane-wave interferers. The estimated eigenvalues are ordered,

$$\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \hat{\lambda}_{min}. \quad (7.119)$$

The eigenvalues can be written as

$$\hat{\lambda}_i = \begin{cases} \hat{\lambda}_i^I + \hat{\lambda}_{min}, & i = 1, 2, \dots, D \\ \hat{\lambda}_i^\epsilon + \hat{\lambda}_{min}, & i = D+1, D+2, \dots, N-1 \\ \hat{\lambda}_{min}, & i = N, \end{cases} \quad (7.120)$$

<sup>6</sup>This example is adapted from Carlson [Car88].

<sup>7</sup>We discuss the behavior for  $\mathbf{S}_n$  and MVDR beamformers. A similar discussion follows for  $\mathbf{S}_x$  and MPDR beamformers.

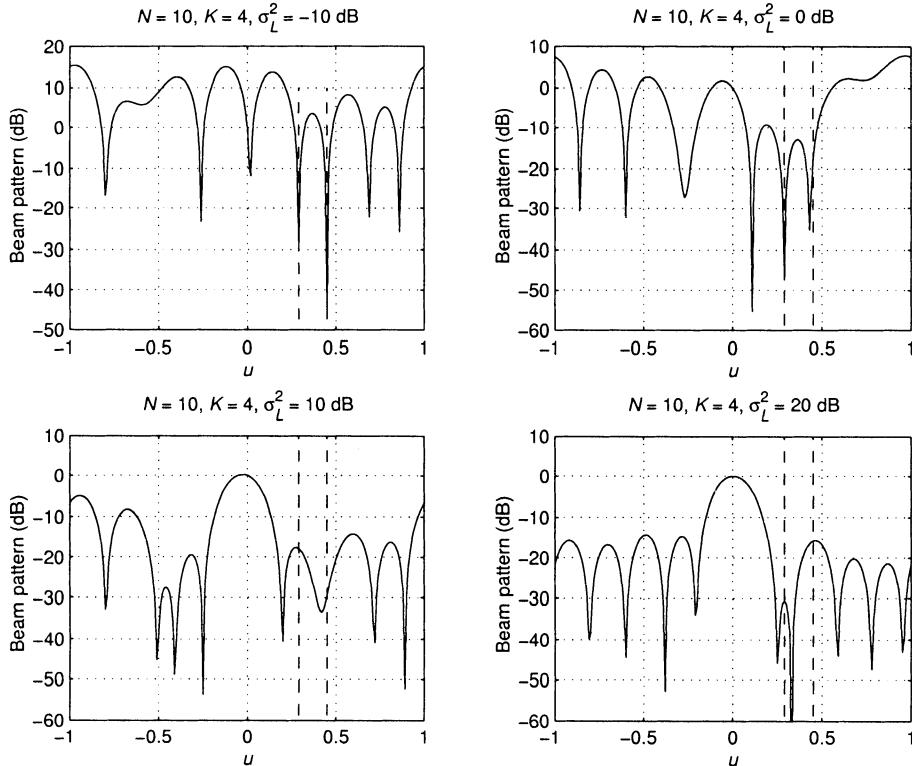


Figure 7.11 MPDR-FL-SMI beamformer: beam pattern for two interferers,  $K = 4$ : (a)  $LNR = -10 \text{ dB}$ ; (b)  $LNR = 0 \text{ dB}$ ; (c)  $LNR = 10 \text{ dB}$ ; (d)  $LNR = 20 \text{ dB}$ .

where  $\hat{\lambda}_{min}$  is the smallest (minimum eigenvalue) of  $\hat{\mathbf{S}}_n$ ;  $\hat{\lambda}_i^\epsilon$  is the difference between the other  $N - D - 1$  noise eigenvalues and  $\hat{\lambda}_{min}$ ; and  $\hat{\lambda}_i^I$  is the estimate of the interference eigenvalues. Note that if  $\hat{\mathbf{S}}_n$  equaled  $\mathbf{S}_n$ , then the  $N - D$  noise eigenvalues would be equal and  $\hat{\lambda}_i^\epsilon$  would be zero.

The inverse of  $\hat{\mathbf{S}}_n^{-1}$  can be written as

$$\begin{aligned} \hat{\mathbf{S}}_n^{-1} &= \sum_{i=1}^D \frac{1}{\hat{\lambda}_i^I + \hat{\lambda}_{min}} \Phi_i \Phi_i^H + \sum_{i=D+1}^{N-1} \frac{1}{\hat{\lambda}_i^\epsilon + \hat{\lambda}_{min}} \Phi_i \Phi_i^H \\ &\quad + \frac{1}{\hat{\lambda}_{min}} \Phi_i \Phi_i^H, \end{aligned} \quad (7.121)$$

or

$$\hat{\mathbf{S}}_n^{-1} = \frac{1}{\hat{\lambda}_{min}} \left\{ \mathbf{I} - \sum_{i=1}^N \frac{\hat{\lambda}_i - \hat{\lambda}_{min}}{\hat{\lambda}_i^R} \Phi_i \Phi_i^H \right\}. \quad (7.122)$$

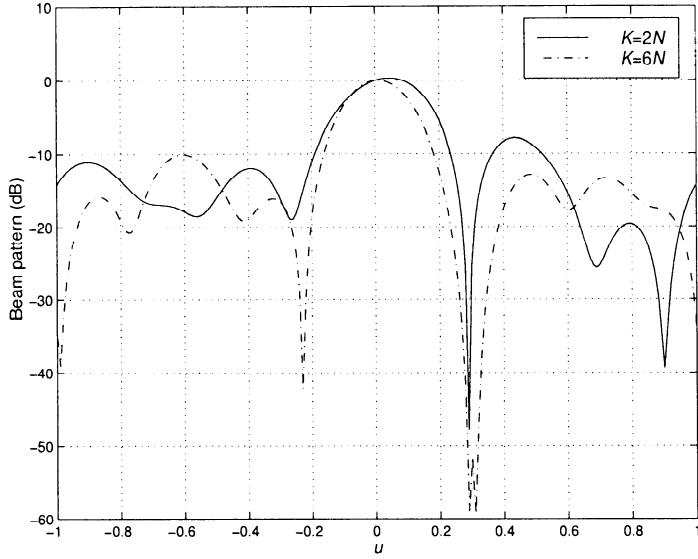


Figure 7.12 MPDR-SMI beamformer with Dolph-Chebychev quiescent beam pattern; adapted beam pattern with  $2N$  and  $6N$  samples:  $u_s = 0$ ,  $u_I = 0.29$ ,  $INR = 30$  dB.

Using (7.120) in (7.83), the optimum weight vector can be written as

$$\mathbf{w}_o^H = \mathbf{w}_q^H - \sum_{i=1}^N \frac{\hat{\lambda}_i - \hat{\lambda}_{min}}{\hat{\lambda}_i} \mathbf{w}_q^H \Phi_i \Phi_i^H, \quad (7.123)$$

where  $\mathbf{w}_q$  is the weight vector in a white noise, or quiescent, environment.<sup>8</sup>

The resulting beam pattern is

$$\begin{aligned} B_o(\psi) &= B_q(\psi) - \sum_{i=1}^N \frac{\hat{\lambda}_i - \hat{\lambda}_{min}}{\hat{\lambda}_i} \rho_{qi} B_{eig,i}(\psi) \\ &= B_q(\psi) - \sum_{i=1}^D \frac{\hat{\lambda}_i^I}{\hat{\lambda}_i} \rho_{qi} B_{eig,i}(\psi) \\ &\quad - \sum_{i=D+1}^{N-1} \frac{\hat{\lambda}_i^\epsilon}{\hat{\lambda}_i^\epsilon + \hat{\lambda}_{min}} \rho_{qi} B_{eig,i}(\psi), \end{aligned} \quad (7.124)$$

where

$$\rho_{qi} = \mathbf{w}_q^H \Phi_i, \quad (7.125)$$

---

<sup>8</sup>We have not normalized  $\mathbf{w}_q$  and  $\mathbf{w}_o$ , in order to simplify the notation.

is the correlation between the quiescent weight vector and the  $i$ th eigenvector and,

$$B_{eig,i}(\psi) = \Phi_i^H \mathbf{v}(\psi), \quad (7.126)$$

is the  $i$ th eigenbeam. Recall that the eigenbeams are orthogonal.

The  $\rho_{qi}$  term in (7.124) scales the interference eigenbeam to be equal to the quiescent beam in the direction of the interference. Therefore, the value of  $B_o(\psi_{Ii})$ ,  $i = 1, \dots, D$ , can be written as

$$\left(1 - \frac{\hat{\lambda}_i - \hat{\lambda}_{min}}{\hat{\lambda}_i}\right)^2 = \left(\frac{\hat{\lambda}_{min}}{\hat{\lambda}_i}\right)^2. \quad (7.127)$$

Therefore, the beam pattern in the direction of the interference is equal to twice the eigenvalue spread in dB.

For large eigenvalues, this term approaches zero so the interferers are almost completely nulled, which is the desired behavior. However, the noise eigenvalues will cause a problem.

Assuming there are  $D$  interferers, then there are  $N - D$  eigenvalues corresponding to noise. If we knew their values exactly, then they would all be equal,

$$\lambda_i = \lambda_{i+1} = \dots = \lambda_D = \lambda_{min}, \quad i > N - D. \quad (7.128)$$

Then, the spread in (7.127) would be unity and the white noise would have no impact on the quiescent pattern, which is the desired behavior. However, we are estimating the  $\lambda_i$  from the sample spectral matrix and, for small  $K$ , there will be a significant spread around the correct value. This means that the beamformer will be subtracting noise eigenbeams (which are random in nature) from the quiescent beam pattern. From (7.124), we see that the noise eigenbeams would not be added if  $\hat{\lambda}_i^c$  were zero. We show this behavior for the model in Example 7.3.5. Note that this is an MPDR example.

#### Example 7.3.6 (continuation)

In Figure 7.13, we show the noise eigenvalue spread as a function of the number of snapshots. We see that there is significant improvement between  $K = N$  and  $K = 2N$ , but the subsequent improvement is gradual.

In Figure 7.14, we show the highest sidelobe in the adapted pattern as a function of the number of snapshots. The level is similar to that predicted by (7.118) (which corresponded to the average level).

In Figure 7.15, we show the reduction in array gain as a function of the number of snapshots. This behavior is consistent with the result in (7.110).

This behavior leads to the conclusion that the noise eigenvalues and eigenvectors can create a significant problem. Once again, we use diagonal loading.

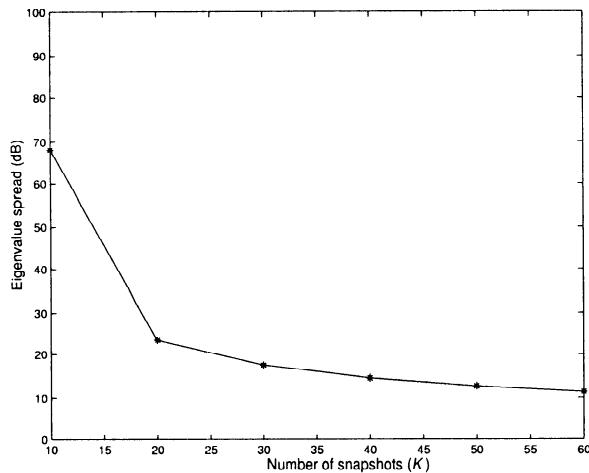


Figure 7.13 Noise eigenvalue spread (ratio of largest noise eigenvalue to smallest) as a function of the number of independent snapshots included in the spectral matrix;  $SNR = 20$  dB,  $u_I = 0.29$ ,  $INR = 30$  dB, eight noise eigenvalues.

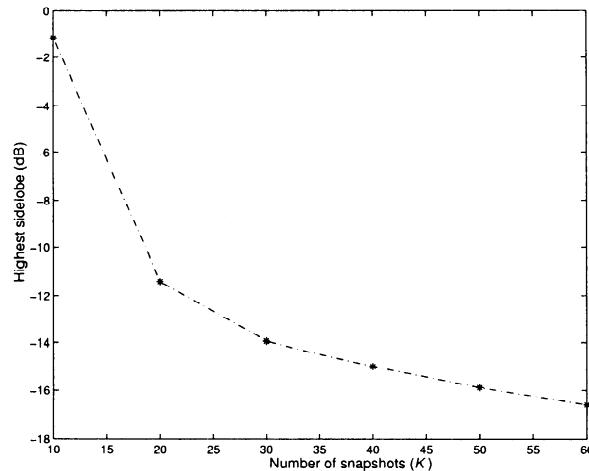


Figure 7.14 MPDR-SMI beamformers: highest sidelobe in adapted patterns as a function of the number of independent snapshots included in the spectral matrix. Dolph-Chebychev (-30 dB SLL) quiescent beam,  $SNR = 20$  dB,  $INR = 30$  dB.

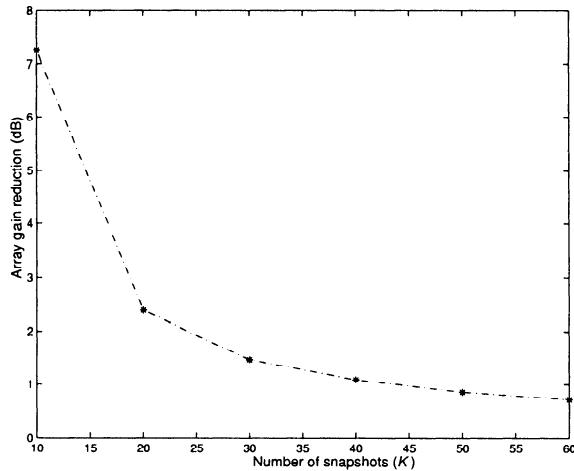


Figure 7.15 MPDR-SMI beamformer: array gain reduction in adapted main beam as a function of the number of independent snapshots included in the covariance matrix.

If we use (7.115) to construct the weight vector, then the coefficient on the noise eigenbeams in (7.124) is

$$\frac{\hat{\lambda}_i^\epsilon}{\hat{\lambda}_i^\epsilon + \hat{\lambda}_{min} + \sigma_L^2}. \quad (7.129)$$

Thus, by choosing a  $\sigma_L^2$  greater than  $\hat{\lambda}_{min}$  (e.g., 10 dB) we can essentially eliminate the noise eigenbeams.

The technique is simple. We add a diagonal matrix  $\sigma_L^2 \mathbf{I}$  to  $\hat{\mathbf{S}}_n$  or  $\hat{\mathbf{S}}_x$  before we utilize it to construct the weighting vector. We illustrate typical behavior with a simple example.

#### Example 7.3.7 (continuation)

Consider a SLA 10. The signal arrives from  $u_s = 0$  with an  $SNR = 20$  dB. The quiescent beam pattern is a Dolph-Chebychev pattern with -30-dB SLL. The single interfering signal arrives from  $u_I = 0.29$  with an  $INR = 30$  dB. The  $LNR$  ( $\triangleq \sigma_L^2/\sigma_w^2$ ) is 10 dB. In Figure 7.16, we show representative beam patterns for  $K = 2N$  and  $K = 6N$  with and without loading. We see that the sidelobe behavior is improved significantly.

In Figure 7.17, we show the noise eigenvalue spread as a function of  $\sigma_L^2/\sigma_w^2$  for various values of  $K$ . In Figure 7.18, we show the highest sidelobes in the adapted pattern as a function of  $\sigma_L^2/\sigma_w^2$  for various values of  $K$ . In Figure 7.19, we show the reduction in array gain as a function of  $\sigma_L^2/\sigma_w^2$  for various values of  $K$ . We see that, for all of the performance metrics of interest, loading offers significant improvement.

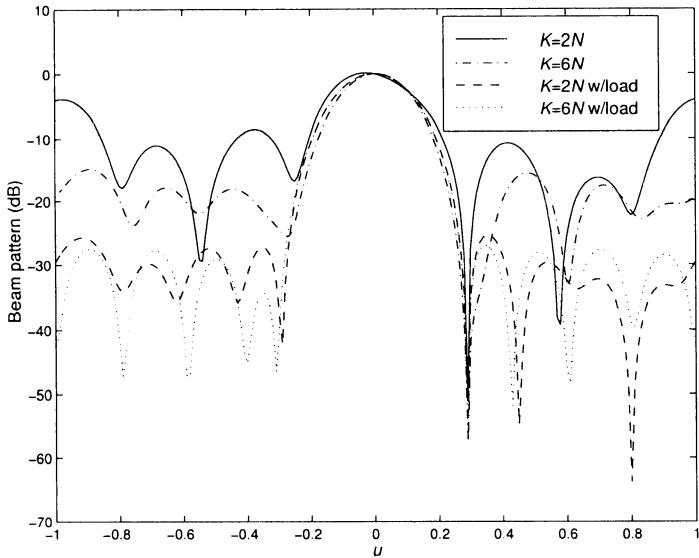


Figure 7.16 MPDR-SMI beamformer: adapted beam pattern with  $6N$  snapshots and diagonal loading;  $u_s = 0$ ,  $SNR = 20$  dB,  $u_I = 0.29$ ,  $INR = 30$  dB,  $LNR = 10$  dB.

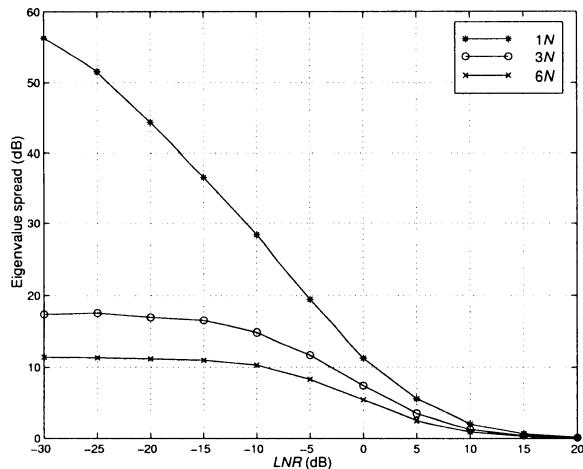


Figure 7.17 MPDR-SMI beamformer: noise eigenvalue spread (ratio of largest noise eigenvalue to smallest) as a function of the loading level for  $1N$ ,  $3N$ , and  $6N$  independent snapshots in spectral matrix;  $u_s = 0$ ,  $SNR = 20$  dB,  $u_I = 0.29$ ,  $INR = 30$  dB.

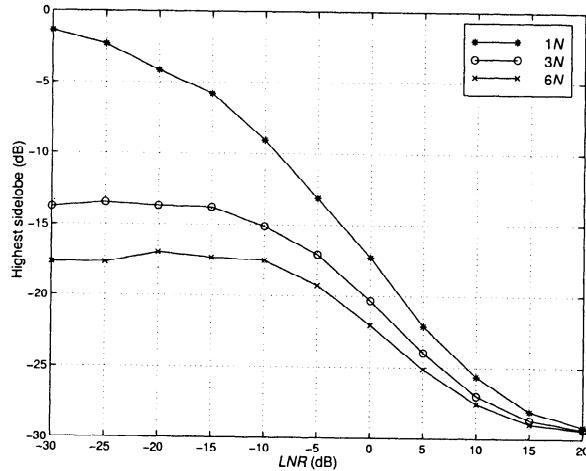


Figure 7.18 MPDR-SMI beamformer: highest sidelobes in adapted pattern as function of loading level for  $1N$ ,  $3N$ , and  $6N$  independent snapshots in spectral matrix.

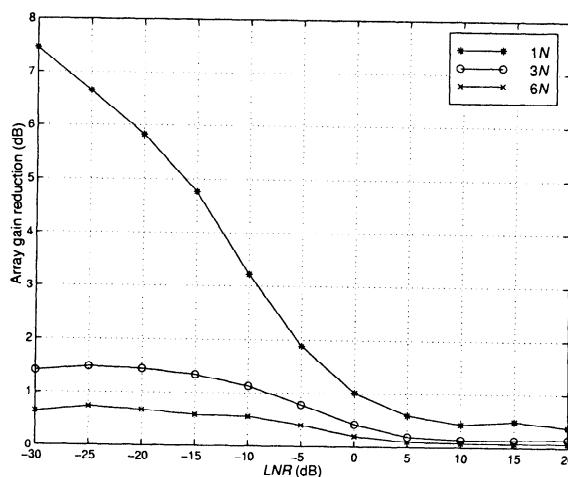


Figure 7.19 MPDR-SMI beamformer: array gain reduction as a function of loading level for  $1N$ ,  $3N$ , and  $6N$  independent snapshots in spectral matrix.

The disadvantage is that the beamformer loses its ability to adapt against small eigenvalues. These can occur if there are small power interferers or if there are two or more large interferers close together. From Figure 5.14, we see that as the interferer separation becomes smaller than the  $BW_{NN}$ , the second (and higher) eigenvalues become quite small. Depending on the scenario, this may or may not be a problem.

Because of its simplicity and potential performance improvement, diagonal loading is an attractive modification to the SMI algorithm in most applications. The disadvantage of fixed diagonal loading is that we need to have prior knowledge of the signal and interferer levels in order to select the appropriate value for  $\sigma_L^2$ . In many applications of interest we have enough prior information about the anticipated environment to choose an appropriate loading level.

Another technique for dealing with the errors in estimating the noise eigenvalues is to replace the estimates of the  $(N - D)$  smallest eigenvalues by their average value,

$$\alpha \triangleq \frac{1}{N - D} \sum_{i=D+1}^N \hat{\lambda}_i. \quad (7.130)$$

This approach is used in the dominant mode rejection (DMR) beamformer that was discussed in Section 6.8.3. We study its adaptive behavior in Section 7.9.

### 7.3.4 Toeplitz Estimators

In Section 7.2.4, we indicated that there are techniques for estimating  $\mathbf{S}_x$  that exploit the Toeplitz structure of  $\mathbf{S}_x$ . Fuhrmann [Fuh91] has shown that these techniques lead to a significant performance improvement in the SMI beamformer. The reader is referred to this reference for a discussion of the technique.

### 7.3.5 Summary

In this section, we have studied the SMI implementation of the MVDR and MPDR beamformer.

For MVDR beamformers, we defined a random variable  $\rho$ , which is the ratio of the  $SINR_{smi}$  to the steady state  $SINR_{mvdr}$ . The probability density of  $\rho$  does not depend on the interference environment. For  $\hat{\mathbf{S}}_n = \mathbf{C}_x$ ,  $E[\rho]$  was  $-1$  dB for  $K = 4N$ . If we use FB averaging,  $E[\rho] = -1$  dB for  $K = 2N$  and we effectively double our sample support.

The performance can be improved by using diagonal loading. This technique, which consists of adding a diagonal matrix to either  $\hat{\mathbf{S}}_x$  or  $\hat{\mathbf{S}}_n$ , works best in a strong interferer environment. Diagonal loading reduces the side-lobe level, improves the  $SINR_{smi}$  performance, and allows  $K < N$  sample support. Diagonal loading plays an important role in most beamformer implementations.

Similar results were obtained for the MPDR beamformer. However, in the presence of a strong signal the convergence was much slower. Diagonal loading provided more dramatic improvements for the MPDR model.

The SMI technique is effective in many applications. However, its computational requirement is a function of  $N^3$ , and it is a block algorithm. In the next section, we develop a recursive algorithm.

## 7.4 Recursive Least Squares (RLS)

In this section, we develop a recursive implementation of the sample matrix inversion algorithm. In order to do this efficiently, we first reformulate the MPDR beamformer and the MMSE beamformer as least squares problems. We do that in Section 7.4.1.

In Section 7.4.2, we develop a recursive algorithm for the MPDR beamformer. In Section 7.4.3 we develop a recursive algorithm for the least squares estimation (LSE) beamformer. Our discussion is adapted from the development in Chapter 13 of [Hay96].<sup>9</sup>

We should observe that the recursive algorithms that we develop in this section are adequate for our present purposes. In Section 7.5, we revisit the implementation problem and develop algorithms that are more computationally efficient and have better numerical stability.

### 7.4.1 Least Squares Formulation

The method of least squares estimation was invented by Gauss in 1795 in his studies of motion of heavenly bodies.<sup>10</sup> In 1912, Fisher [Fis12] introduced the maximum likelihood method, which leads to the same result if the observation noise is assumed to be Gaussian. Kolmogorov [Kol41a], [Kol41b] in 1941 and Wiener [Wie49] independently invented MMSE filter theory that

---

<sup>9</sup>An alternative method to develop the RLS algorithm is start with a deterministic Newton-type algorithm and find the stochastic version of it. We discuss this approach briefly in Section 7.7.

<sup>10</sup>Sorenson [Sor70] has provided a good discussion of the history of LSE and its evolution from Gauss to Swerling and Kalman. Our historical discussion is based on this reference.

we have studied in DEMT I [VT68], [VT01a]. In our study of the MMSE filter we utilized ensemble averages. By contrast, LSE utilizes time averages. We find that the least squares beamformers have the same form as the MPDR and MMSE beamformers, with the ensemble averages replaced by time averages.

In the middle 1950s, there was a sequence of papers dealing with recursive implementations of LSEs. These include Follin [CF56], Swerling [Swe58], [Swe59], Kalman and Bucy [KB61]. The culmination was the discrete time Kalman filter [Kal60], which has been applied to a wide range of applications in the last four decades. As Sorenson points out, Swerling's recursive algorithm predates Kalman's work and is essentially the same algorithm. We now derive the least squares estimator.

The output of a distortionless response beamformer is

$$Y(k) = D(k) + N(k), \quad k = 1, 2, \dots, K, \quad (7.131)$$

where  $D(k)$  is the desired signal. The estimation error is  $N(k)$ . In the least squares approach, we minimize a weighted summation of the squared error

$$\xi_N(K) = \sum_{k=1}^K \mu^{K-k} |N(k)|^2, \quad (7.132)$$

where  $\mu$  is a positive constant less than 1. It provides an exponential weighting factor of the error so that the importance of past errors decrease as their distance from the current sample time  $K$  increases. This factor allows the beamformer to accommodate possible non-stationarities in the input. Typically  $\mu$  is close to unity.

Due to the distortionless constraint, the minimization of  $\xi_N(K)$  is equivalent to minimizing

$$\xi_Y(K) = \sum_{k=1}^K \mu^{K-k} |Y(k)|^2. \quad (7.133)$$

As in Chapter 6,

$$Y(k) = \mathbf{w}^H(K) \mathbf{X}(k), \quad (7.134)$$

and the distortionless constraint is

$$\mathbf{w}^H(K) \mathbf{v}_s = 1. \quad (7.135)$$

Note that  $\mathbf{w}^H(K)$  is a function of  $K$  because it will adapt as we receive more data. We minimize  $\xi_Y(K)$  subject to the constraint in (7.135). Define

$$F \triangleq \sum_{k=1}^K \mu^{K-k} \mathbf{w}^H(K) \mathbf{X}(k) \mathbf{X}^H(k) \mathbf{w}(K)$$

$$+ \lambda \left[ \mathbf{w}^H(K) \mathbf{v}_s - 1 \right] + \lambda^* \left[ \mathbf{v}_s^H \mathbf{w}(K) - 1 \right], \quad (7.136)$$

or

$$F = \mathbf{w}^H(K) \Phi(K) \mathbf{w}(K) + \lambda \left[ \mathbf{w}^H(K) \mathbf{v}_s - 1 + \mathbf{v}_s^H \mathbf{w}(K) - 1 \right], \quad (7.137)$$

where

$$\Phi(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.138)$$

The matrix  $\Phi(K)$  is introduced to denote the exponential weighted sample spectral matrix, in contrast with the conventional sample spectral matrix  $\mathbf{C}_x(K)$ .

Substituting (7.138) into (7.136) gives

$$F = \mathbf{w}^H(K) \Phi(K) \mathbf{w}(K) + \lambda \left[ \mathbf{w}^H(K) \mathbf{v}_s - 1 \right] + \lambda^* \left[ \mathbf{v}_s^H \mathbf{w}(K) - 1 \right]. \quad (7.139)$$

Taking the complex gradient with respect to  $\mathbf{w}^H(K)$ , setting the result to zero, and solving for the Lagrange multiplier gives

$$\hat{\mathbf{w}}_{mpdr}(K) = \frac{\Phi^{-1}(K) \mathbf{v}_s}{\mathbf{v}_s^H \Phi^{-1}(K) \mathbf{v}_s} = \Lambda(K) \Phi^{-1}(K) \mathbf{v}_s, \quad (7.140)$$

where

$$\Lambda(K) \triangleq \left[ \mathbf{v}_s^H \Phi^{-1}(K) \mathbf{v}_s \right]^{-1}. \quad (7.141)$$

Note that

$$\xi_Y(K) = \left[ \mathbf{v}_s^H \Phi^{-1}(K) \mathbf{v}_s \right]^{-1} = \Lambda(K). \quad (7.142)$$

We see that the least squares distortionless response beamformer is the MPDR beamformer of Chapter 6 with the ensemble average replaced by a weighted time average.

Note that

$$E[\Phi(K)] = \frac{1 - \mu^K}{1 - \mu} \mathbf{S}_x, \quad (7.143)$$

which is approximately

$$E[\Phi(K)] \simeq \frac{1}{1 - \mu} \mathbf{S}_x, \quad (7.144)$$

for large  $K$ , so  $\Phi(K)$  provides an asymptotically biased estimate of  $\mathbf{S}_x$ .

An alternative definition that provides an unbiased estimate is

$$\Phi'(K) = \frac{1 - \mu}{1 - \mu^K} \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k), \quad (7.145)$$

which, for large  $K$ , reduces to

$$\Phi'(K) = (1 - \mu) \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.146)$$

Using (7.145) in (7.140) gives the same result as before because it appears in both the numerator and denominator. However, when we add diagonal loading we have to take the  $(1 - \mu)$  factor into account.

The least squares solution analogous to the MMSE beamformer follows in a similar manner. We assume there is a desired response  $D(k)$ . The error at time  $k$  is

$$e(k) = D(k) - \mathbf{w}^H(K) \mathbf{X}(k), \quad k = 1, \dots, K. \quad (7.147)$$

Note  $\mathbf{w}^H(K)$  is the weight vector at  $K$ . We minimize

$$\begin{aligned} \xi_\mu(K) &= \sum_{k=1}^K \mu^{K-k} |e(k)|^2 \\ &= \sum_{k=1}^K \mu^{K-k} (D(k) - \mathbf{w}^H(K) \mathbf{X}(k)) (D(k)^* - \mathbf{X}^H(k) \mathbf{w}(K)). \end{aligned} \quad (7.148)$$

Taking the gradient with respect to  $\mathbf{w}^H(K)$  and setting the result to zero, gives

$$\boxed{\hat{\mathbf{w}}_{lse}(K) = \Phi^{-1}(K) \Phi_{xd^*}(K)}, \quad (7.149)$$

where  $\Phi(K)$  is defined in (7.138) and

$$\boxed{\Phi_{xd^*}(K) \triangleq \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) D^*(k)}. \quad (7.150)$$

This result is the MMSE beamformer of Chapter 6, with the ensemble averages replaced by weighted time averages.

The output  $Y(K)$  is

$$Y(K) = \hat{\mathbf{w}}_{lse}^H(K) \mathbf{X}(K). \quad (7.151)$$

In order to implement (7.149), we must generate  $\Phi_{xd^*}(K)$ . We discuss techniques for doing this later in this section.

### 7.4.2 Recursive Implementation

In order to implement  $\hat{\mathbf{w}}_{mpdr}(K)$  recursively, we need an algorithm to find  $\Phi^{-1}(K)$  from  $\Phi^{-1}(K - 1)$ . From (7.138),

$$\Phi(K) = \mu\Phi(K - 1) + \mathbf{X}(K)\mathbf{X}^H(K). \quad (7.152)$$

The desired iteration follows directly from the matrix inversion formula in (A.49)<sup>11</sup>

$$\Phi^{-1}(K) = \mu^{-1}\Phi^{-1}(K - 1) - \frac{\mu^{-2}\Phi^{-1}(K - 1)\mathbf{X}(K)\mathbf{X}^H(K)\Phi^{-1}(K - 1)}{1 + \mu^{-1}\mathbf{X}^H(K)\Phi^{-1}(K - 1)\mathbf{X}(K)}. \quad (7.153)$$

We now define

$$\mathbf{P}(K) = \Phi^{-1}(K), \quad (7.154)$$

and

$$\mathbf{g}(K) = \frac{\mu^{-1}\mathbf{P}(K - 1)\mathbf{X}(K)}{1 + \mu^{-1}\mathbf{X}^H(K)\mathbf{P}(K - 1)\mathbf{X}(K)}. \quad (7.155)$$

The choice of notation is deliberate because of the relationship between recursive least squares and Kalman filtering.

Using (7.154) and (7.155) in (7.153) gives

$$\mathbf{P}(K) = \mu^{-1}\mathbf{P}(K - 1) - \mu^{-1}\mathbf{g}(K)\mathbf{X}^H(K)\mathbf{P}(K - 1), \quad (7.156)$$

which is known as the Riccati equation.

Post-multiplying both sides of (7.156) by  $\mathbf{X}(K)$  and using (7.155), one can show that

$$\mathbf{g}(K) = \mathbf{P}(K)\mathbf{X}(K) = \Phi^{-1}(K)\mathbf{X}(K). \quad (7.157)$$

The vector  $\mathbf{g}(K)$  is referred as the gain vector.

We now develop a recursive equation for  $\hat{\mathbf{w}}_{mpdr}(K)$ . Suppressing the subscript on  $\hat{\mathbf{w}}(K)$ , (7.140) can be written as

$$\hat{\mathbf{w}}(K) = \Lambda(K)\mathbf{P}(K)\mathbf{v}_s. \quad (7.158)$$

Using (7.156) in (7.158) gives

$$\begin{aligned} \hat{\mathbf{w}}(K) &= \Lambda(K) \left[ \mu^{-1}\mathbf{P}(K - 1) - \mu^{-1}\mathbf{g}(K)\mathbf{X}^H(K)\mathbf{P}(K - 1) \right] \mathbf{v}_s \\ &= \left\{ \frac{\Lambda(K)}{\mu\Lambda(K - 1)} \left[ \mathbf{I} - \mathbf{g}(K)\mathbf{X}^H(K) \right] \right\} \hat{\mathbf{w}}(K - 1). \end{aligned} \quad (7.159)$$

---

<sup>11</sup>This approach is due originally to Baird [Bai74]. It is discussed in Hudson's book [Hud81], pp. 124–125, Compton's book [Com88], pp. 318–326, and Haykin's book [Hay96], pp. 566–571.

The term in curly brackets is an  $N \times N$  matrix used to update  $\hat{\mathbf{w}}(K - 1)$ .

The last step is to initialize the algorithm. Haykin [Hay96] suggests augmenting  $\Phi(K)$  with a white noise term. Thus,

$$\Phi(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k) + \sigma_o^2 \mu^K \mathbf{I}, \quad (7.160)$$

where  $\sigma_o^2$  is a small positive constant. This augmentation is just exponentially decaying diagonal loading. Then, for  $K = 0$  (no snapshots received)

$$\Phi(0) = \sigma_o^2 \mathbf{I}, \quad (7.161)$$

and

$$\mathbf{P}(0) = \frac{1}{\sigma_o^2} \mathbf{I}. \quad (7.162)$$

We can choose  $\hat{\mathbf{w}}(0)$  to be any vector satisfying the distortionless constraint. We refer to this vector as the **quiescent** weight vector,  $\mathbf{w}_q$ .

In the time-domain case (equalizers), it is normal to make  $\sigma_o^2$  very small and let  $\hat{\mathbf{w}}(0) = \mathbf{0}$ . However, a small  $\sigma_o^2$  gives very poor beam patterns (distorted main lobes and high sidelobes) for small  $K$ . If we assume the array is in operation prior to adaption, then  $\hat{\mathbf{w}}(0) = \mathbf{0}$  is not a useful initial weighting. We normally initialize the algorithm with

$$\hat{\mathbf{w}}(0) = \frac{\mathbf{v}_s}{N}, \quad (7.163)$$

or a  $\mathbf{w}_q$  with a better sidelobe pattern that satisfies the distortionless criterion.

The effect of the diagonal loading term,  $\sigma_o^2 \mathbf{I}$ , decays rapidly. Taking the expectation of (7.160) gives

$$E[\Phi(K)] = \left[ \frac{1 - \mu^K}{1 - \mu} \right] \mathbf{S}_{\mathbf{x}} + \mu^K \sigma_o^2 \mathbf{I}. \quad (7.164)$$

Defining,

$$\tilde{\Phi}(K) = \left[ \frac{1 - \mu}{1 - \mu^K} \right] \Phi(K), \quad (7.165)$$

(7.164) can be rewritten as

$$E[\tilde{\Phi}(K)] = \mathbf{S}_{\mathbf{x}} + \left[ \frac{1 - \mu}{1 - \mu^K} \right] \mu^K \sigma_o^2 \mathbf{I}. \quad (7.166)$$

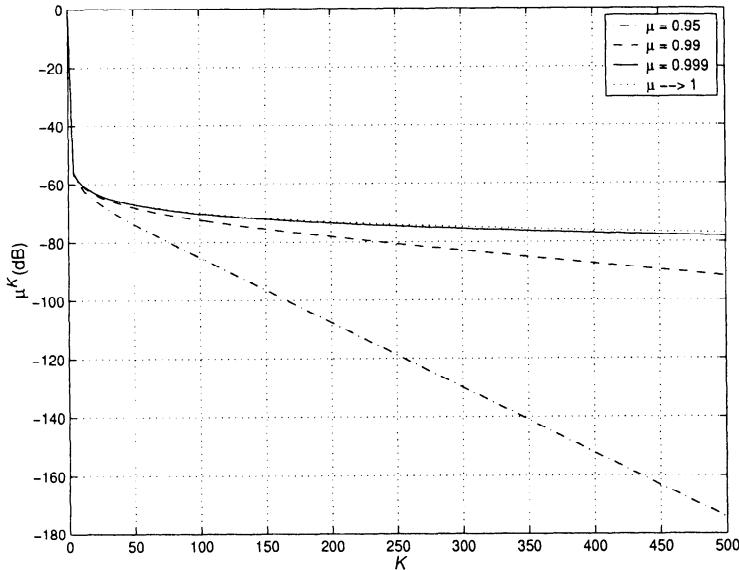


Figure 7.20 Diagonal loading decay in RLS algorithm.

In Figure 7.20, we plot the diagonal loading term as a function of  $K$ . Note that as  $\mu$  approaches unity, we can approximate the fraction as

$$\left[ \frac{1 - \mu}{1 - \mu^K} \right] \simeq \frac{1}{K - 1}. \quad (7.167)$$

Because of this rapid decay, we may have to add additional diagonal loading to maintain robustness. We will revisit that issue later.

Sayed and Kailath [SK94] pointed out that, with this choice of  $\mathbf{P}(0)$ , we are actually minimizing

$$\min_{\mathbf{w}(K)} \left[ \sigma_o^2 \mu^K \| \mathbf{w}(K) \|^2 + \sum_{k=1}^K \mu^{K-k} |Y(k)|^2 \right], \quad (7.168)$$

where  $Y(k)$  is given by (7.134). Note that the first term is the norm of the weight vector multiplied by an exponentially decreasing constant. We will encounter the first term in various contexts.

The steps can be summarized as:

Initialize the algorithm with

$$\mathbf{P}(0) = \frac{1}{\sigma_o^2} \mathbf{I}, \quad (7.169)$$

$$\hat{\mathbf{w}}(0) = \frac{\mathbf{v}_s}{N}, \quad (7.170)$$

or a  $\mathbf{w}_q$  with a better sidelobe pattern that satisfies the distortionless criterion. Note that (7.169) provides the initialization for  $\Lambda(0)$  by using (7.169) in (7.173).

At each snapshot,  $K = 1, 2, \dots$ , compute

$$\mathbf{g}(K) = \frac{\mu^{-1} \mathbf{P}(K-1) \mathbf{X}(K)}{1 + \mu^{-1} \mathbf{X}^H(K) \mathbf{P}(K-1) \mathbf{X}(K)}, \quad (7.171)$$

$$\mathbf{P}(K) = \mu^{-1} \mathbf{P}(K-1) - \mu^{-1} \mathbf{g}(K) \mathbf{X}^H(K) \mathbf{P}(K-1), \quad (7.172)$$

$$\Lambda(K) = [\mathbf{v}_s^H \mathbf{P}(K) \mathbf{v}_s]^{-1}, \quad (7.173)$$

and

$$\hat{\mathbf{w}}_{mpdr}(K) = \frac{\Lambda(K)}{\mu \Lambda(K-1)} [\mathbf{I} - \mathbf{g}(K) \mathbf{X}^H(K)] \hat{\mathbf{w}}_{mpdr}(K-1). \quad (7.174)$$

Then,

$$Y(K) = \hat{\mathbf{w}}_{mpdr}^H(K) \mathbf{X}(K). \quad (7.175)$$

We can also write (7.174) as

$$\hat{\mathbf{w}}_{mpdr}(K) = \frac{\Lambda(K)}{\mu \Lambda(K-1)} [\hat{\mathbf{w}}_{mpdr}(K-1) - \mathbf{g}(K) \tilde{Y}^*(K)], \quad (7.176)$$

where

$$\tilde{Y}(K) \triangleq \hat{\mathbf{w}}_{mpdr}^H(K-1) \mathbf{X}(K). \quad (7.177)$$

The expressions in (7.174) and (7.176) are identical in the absence of numerical errors. In practice, we normally use (7.176) so that the RLS algorithm is operating as a closed-loop system, as shown in Figure 7.21. This implementation is due to Baird [Bai73] (e.g., [Hud79] and p.127 in [Hud81]).

We consider the following example to illustrate the behavior.

#### Example 7.4.1 (continuation, Examples 6.3.2–6.3.5, 7.3.2)

Consider a standard 10-element linear array in which the desired signal arrives from  $u_s = 0$ , and two equal-power interfering signals arrive from  $u_I = 0.29$  and  $0.45$  with an *INR* = 20 dB. We use a recursive LSE to implement the MPDR beamformer ((7.169)–(7.177)).

In Figure 7.22, we plot the average output *SINR*<sub>o</sub> versus  $K$  for  $\mu = 0.99$  and  $\mu = 0.999$ . We also show the SMI beamformer performance with no diagonal loading for comparison. In Figure 7.22(a), the initial loading,  $\sigma_o^2/\sigma_w^2$ , equals -10 dB. For  $\mu = 0.999$ , the behavior of RLS and SMI is essentially equal. For  $\mu = 0.99$ , the RLS performance levels off because we are effectively using about  $((1 - \mu)^{-1} = 100)$  snapshots. In Figure 7.22(b), the initial

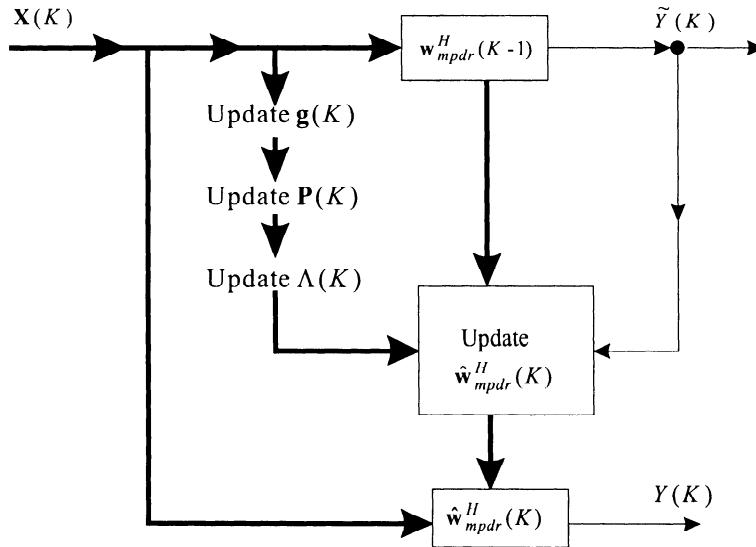


Figure 7.21 MPDR-RLS algorithm: closed-loop implementation.

loading is 0 dB. The increased loading causes RLS to be better for  $K < 40$ . For larger  $K$ , it has decayed and the performance of RLS with  $\mu = 0.999$  and SMI are the same. In Figure 7.22(c), the initial loading is 10 dB. Now RLS is better for  $K < 200$ .

In Figure 7.23, we compare the performance for various  $SNR$  with  $\sigma_o^2/\sigma_w^2 = 10$  dB.

We observe that the RLS algorithm does not perform as well as the diagonally loaded SMI algorithm. The reason is that the diagonal loading in the RLS algorithm decreases exponentially. This suggests defining a different augmented matrix,

$$\Phi_a(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k) + \sigma_L^2 \mathbf{I}, \quad (7.178)$$

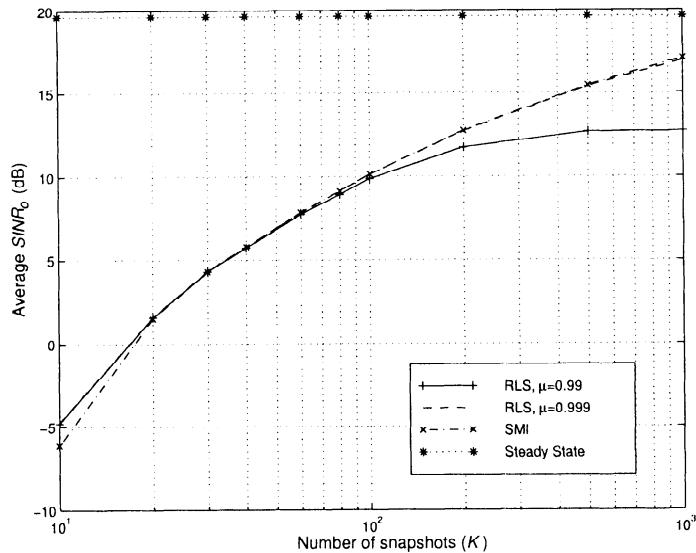
To illustrate the behavior we use an algorithm that accomplishes (7.178), but is not practical to implement in practice.

We replace  $\Phi(K)$  in (7.152) with

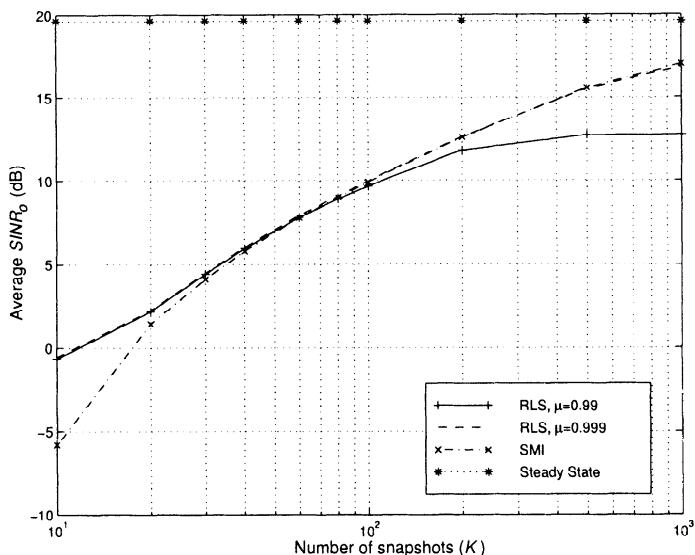
$$\Phi_a(K) = \mu\Phi_a(K-1) + \mathbf{X}(K)\mathbf{X}^H(K) + (1-\mu)\sigma_l^2\mathbf{I}, \quad (7.179)$$

and write (7.156) as

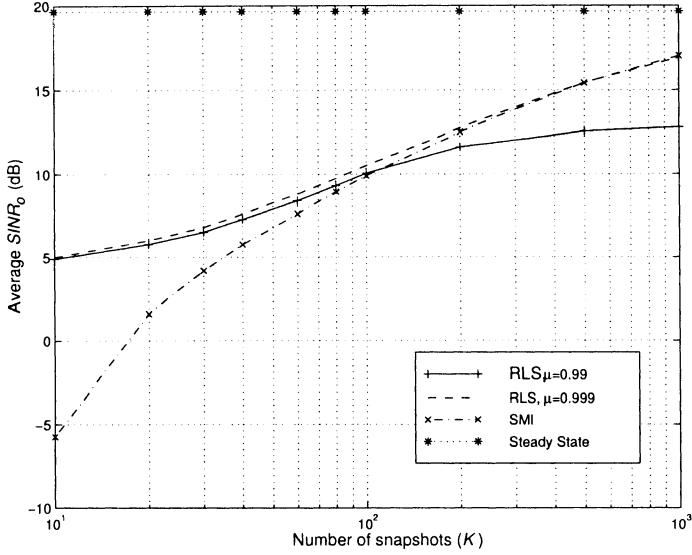
$$\mathbf{P}_1(K) = \mu^{-1} \mathbf{P}(K-1) - \mu^{-1} \mathbf{g}(K) \mathbf{X}^H(K) \mathbf{P}(K-1). \quad (7.180)$$



(a)



(b)



(c)

Figure 7.22 MPDR-SMI and MPDR-RLS beamformers:  $SNR = 10$  dB,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB,  $\mu = 0.99$  and  $0.999$ ;  $\overline{SINR}_o$  versus  $K$ : (a)  $\sigma_o^2/\sigma_w^2 = -10$  dB; (b)  $\sigma_o^2/\sigma_w^2 = 0$  dB; (c)  $\sigma_o^2/\sigma_w^2 = 10$  dB.

Now define  $\mathbf{A}(K)$  as

$$\mathbf{A}(K) = \mathbf{P}_1(K) \left[ \mathbf{P}_1(K) + \frac{1}{\sigma_L^2} \mathbf{I} \right]^{-1}. \quad (7.181)$$

Then,

$$\begin{aligned} \mathbf{P}(K) &= \mathbf{P}_1 - \mathbf{A}(K)\mathbf{P}_1(K) \\ &= \mu^{-1} [\mathbf{I} - \mathbf{A}(K)] [\mathbf{I} - \mathbf{g}(K)\mathbf{X}^H(K)] \mathbf{P}(K-1), \end{aligned} \quad (7.182)$$

with

$$\hat{\mathbf{w}} = \frac{\mu^{-1}\Lambda(K)}{\Lambda(K-1)} [\mathbf{I} - \mathbf{A}(K)] [\hat{\mathbf{w}}(K-1) - \mathbf{g}(K)\mathbf{X}^H(K)\hat{\mathbf{w}}(K-1)]. \quad (7.183)$$

The algorithm is not practical because the inverse in (7.181) is required at each iteration. However, it provides a performance reference. An alternative

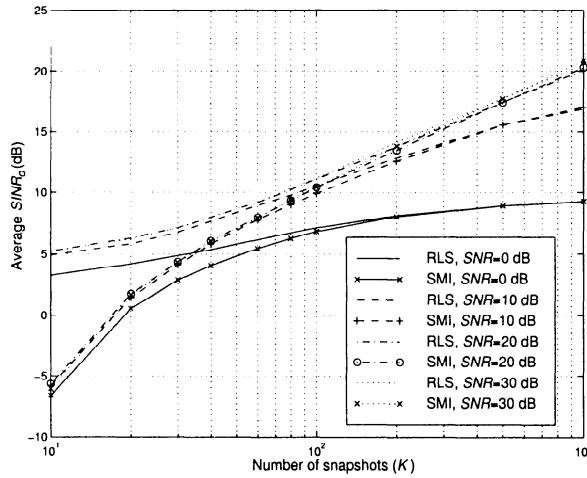


Figure 7.23 MPDR-SMI and MPDR-RLS beamformers:  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB,  $\mu = 0.999$ ,  $\sigma_o^2/\sigma_w^2 = 10$  dB,  $SNR = 0, 10, 20$  dB;  $\overline{SINR}_o$  versus  $K$ .

approach is to use a moving window of length  $1/(1-\mu)$ . The moving window counterpart to (7.178) is easy to update<sup>12</sup> (see Problem 7.4.24).

In the GSC structure, we can utilize an approximate procedure. We develop it in Section 7.4.4.

### 7.4.3 Recursive Implementation of LSE Beamformer

Using a similar approach one can derive a recursive implementation of the LSE beamformer discussed in (7.149) and (7.150). The equations specifying  $\hat{\mathbf{w}}_{lse}(K)$  are

$$\hat{\mathbf{w}}_{lse}(K) = \Phi^{-1}(K)\Phi_{xd^*}(K) = \mathbf{P}(K)\Phi_{xd^*}(K) \quad (7.184)$$

and

$$\begin{aligned} \Phi_{xd^*}(K) &= \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) D^*(k) \\ &= \mathbf{X}(K) D^*(K) + \mu \Phi_{xd^*}(K-1). \end{aligned} \quad (7.185)$$

<sup>12</sup>This alternative was suggested by P. Stoica (private communication).

The results in (7.169)–(7.172) still apply. Using (7.185) and (7.172) in (7.184) gives

$$\hat{\mathbf{w}}_{lse}(K) = \hat{\mathbf{w}}_{lse}(K-1) + \mathbf{g}(K) \left[ D^*(K) - \mathbf{X}^H(K) \hat{\mathbf{w}}_{lse}(K-1) \right]. \quad (7.186)$$

Now define

$$e_p(K) = D(K) - \hat{\mathbf{w}}_{lse}^H(K-1) \mathbf{X}(K). \quad (7.187)$$

The subscript  $p$  in  $e_p(K)$  denotes prior. The quantity  $e_p(K)$  is the error between the desired output  $D(K)$  and the beamformer output when the current input sample  $\mathbf{X}(K)$  is applied to the prior weight vector  $\hat{\mathbf{w}}_{lse}^H(K-1)$ . The prior error can also be written using

$$\tilde{Y}(K) = \hat{\mathbf{w}}_{lse}^H(K-1) \mathbf{X}(K), \quad (7.188)$$

to provide a closed-loop implementation. Using (7.187) in (7.186) gives

$$\hat{\mathbf{w}}_{lse}(K) = \hat{\mathbf{w}}_{lse}(K-1) + \mathbf{g}(K) e_p^*(K), \quad (7.189)$$

and

$$Y(K) = \hat{\mathbf{w}}_{lse}^H(K) \mathbf{X}(K). \quad (7.190)$$

The algorithm can be summarized:

1. Initialize the algorithm with

$$\mathbf{P}(0) = \frac{1}{\sigma_o^2} \mathbf{I}, \quad (7.191)$$

$$\hat{\mathbf{w}}(0) = \frac{\mathbf{v}_s}{N}, \quad (7.192)$$

or a  $\mathbf{w}_q$  with a better sidelobe pattern.

2. At each snapshot,  $K = 1, 2, \dots$ , compute

$$\mathbf{g}(K) = \frac{\mu^{-1} \mathbf{P}(K-1) \mathbf{X}(K)}{1 + \mu^{-1} \mathbf{X}^H(K) \mathbf{P}(K-1) \mathbf{X}(K)}, \quad (7.193)$$

and

$$\mathbf{P}(K) = \mu^{-1} \mathbf{P}(K-1) - \mu^{-1} \mathbf{g}(K) \mathbf{X}^H(K) \mathbf{P}(K-1). \quad (7.194)$$

3. Compute  $e_p(K)$  using (7.187).
4. Compute  $\hat{\mathbf{w}}_{lse}(K)$  using (7.189).

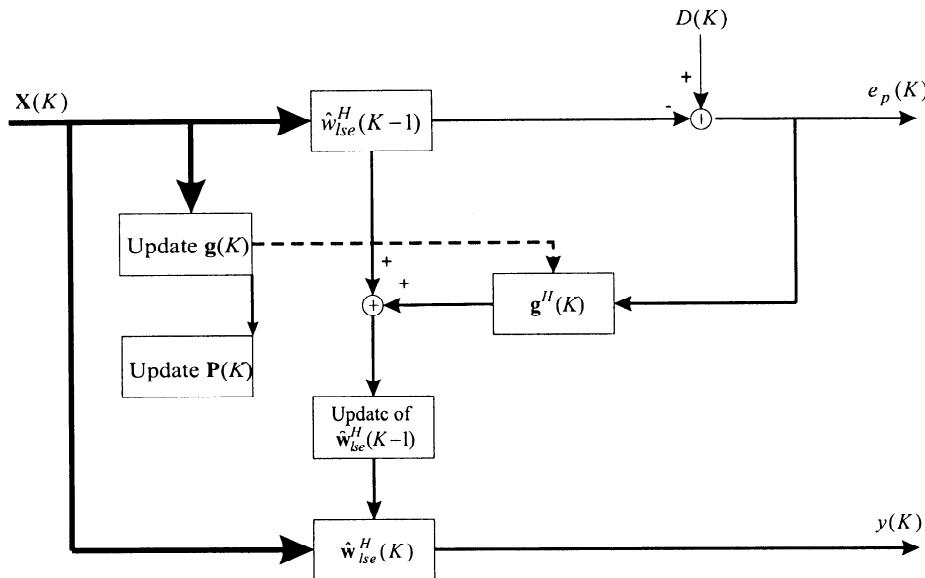


Figure 7.24 Implementation of LSE beamformer.

5. Compute the output  $Y(K)$  using (7.190).

Note that in the direct form implementation of the LSE beamformer, the desired signal  $D(k), k = 1, 2, \dots, K$  must be known at the receiving antenna. In a typical communications system implementation, the desired signal  $D(k)$  could be supplied by a training sequence transmitted at the beginning of a frame. We can then switch to a decision directed mode for continuing operation.

Note that, in contrast to the MVDR and MPDR beamformers, the receiver does not need to know the direction of arrival of the signal. Figure 7.24 shows the implementation of the algorithm.

In our discussion in the text, the emphasis is on MPDR and MVDR beamformers. When we implement these beamformers as generalized side-lobe cancellers (see Section 6.7.3 and Figure 6.46), the LSE beamformer is the adaptive element in the lower path and the desired signal is the output of the quiescent vector processor in the upper path. We discuss this implementation in detail in the next section.

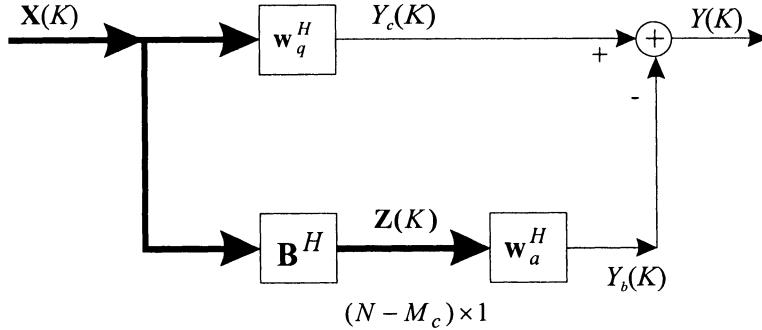


Figure 7.25 Generalized sidelobe canceller.

#### 7.4.4 Generalized Sidelobe Canceller

The generalized sidelobe canceller was shown in Figure 6.46 and is repeated in Figure 7.25 for convenience.

In most applications we will implement  $w_a$  using recursive LSE from Section 7.4.3 or an LMS algorithm that we will discuss in Section 7.7. The adaptive processor  $w_a^H$  is a least squares estimator (Section 7.4.3), with  $Y_c(k)$  as the desired signal and  $Z(k)$ , the output of the blocking matrix, as the input.

From (7.138),

$$\Phi_x(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{X}(k) \mathbf{X}^H(k). \quad (7.195)$$

Then,

$$\Phi_z(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{Z}(k) \mathbf{Z}^H(k) = \mathbf{B}^H \Phi_x(K) \mathbf{B}, \quad (7.196)$$

and

$$\Phi_{zy_c^*}(K) = \sum_{k=1}^K \mu^{K-k} \mathbf{Z}(k) \mathbf{Y}_c^*(k) = \mathbf{B}^H \Phi_x(K) \mathbf{w}_q. \quad (7.197)$$

We define

$$\mathbf{P}_z(K) = \Phi_z^{-1}(K). \quad (7.198)$$

For this case the RLS algorithm is adapted from (7.193)–(7.194) and (7.187)–(7.190):

$$\mathbf{g}_z(K) = \frac{\mu^{-1} \mathbf{P}_z(K-1) \mathbf{Z}(K)}{1 + \mu^{-1} \mathbf{Z}^H(K) \mathbf{P}_z(K-1) \mathbf{Z}(K)}. \quad (7.199)$$

$$\mathbf{P}_z(K) = \mu^{-1} \mathbf{P}_z(K-1) - \mu^{-1} \mathbf{g}_z(K) \mathbf{Z}^H(K) \mathbf{P}_z(K-1). \quad (7.200)$$

$$e_p(K) = Y_c(K) - \hat{\mathbf{w}}_a^H(K-1)\mathbf{Z}(K). \quad (7.201)$$

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K)e_p^*(K). \quad (7.202)$$

The result in (7.202) can also be written as

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K) \left[ Y_c^*(K) - \mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1) \right], \quad (7.203)$$

or

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K) \left[ Y_c^*(K) - \bar{Y}_b^*(K) \right]. \quad (7.204)$$

The initial conditions are

$$\mathbf{P}_z(0) = \frac{1}{\sigma_o^2} \left[ \mathbf{B}^H \mathbf{B} \right]^{-1} = \frac{1}{\sigma_o^2} \mathbf{I}_{(N-M_c)}. \quad (7.205)$$

$$\hat{\mathbf{w}}_a(0) = \mathbf{0}. \quad (7.206)$$

Note that we are operating on an  $N - M_c$  vector which provides a computational advantage.

In order to include fixed loading, note that

$$\mathbf{g}_z(K) = \mathbf{P}_z(K)\mathbf{Z}(K). \quad (7.207)$$

Then we can write

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{P}_z(K)\mathbf{Z}(K)Y_c^*(K) - \mathbf{P}_z(K)\mathbf{Z}(K)\mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1). \quad (7.208)$$

At each iteration, there is a term of the form  $\mathbf{Z}(K)\mathbf{Z}^H(K)$  to which a diagonal matrix can be added. The fixed diagonal loading RLS update becomes

$$\begin{aligned} \hat{\mathbf{w}}_a(K) &= \hat{\mathbf{w}}_a(K-1) + \mathbf{P}_z(K)\mathbf{Z}(K)Y_c^*(K) - \mathbf{P}_z(K) \times \\ &\quad \left[ \mathbf{Z}(K)\mathbf{Z}^H(K) + \sigma_L^2 \mathbf{I} \right] \hat{\mathbf{w}}_a(K-1). \end{aligned} \quad (7.209)$$

This update equation can be written as

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K)e_p^*(K) - \sigma_L^2 \mathbf{P}_z(K)\hat{\mathbf{w}}_a(K-1), \quad (7.210)$$

or

$$\hat{\mathbf{w}}_a(K) = \left[ \mathbf{I} - \sigma_L^2 \mathbf{P}_z(K) \right] \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K)e_p^*(K). \quad (7.211)$$

Note that this approach is ad hoc in that it does not actually add loading to the sample spectral matrix.

We consider two examples to illustrate the performance.

**Example 7.4.2** (continuation, Example 7.4.1)

Consider the same model as in Example 7.4.1. We repeat the simulation using the same data set and parameter values. We incorporate diagonal loading using (7.209). We also show the results using the diagonal loading technique in (7.179)–(7.183) (denoted by *RLS, FLx*). The results are shown in Figure 7.26.

For  $\mu = 0.999$ , the (impractical) fixed loading scheme has the same performance as the diagonally loaded SMI algorithm. The GSC implementation in (7.211) is only slightly worse and is straightforward to realize. For  $\mu = 0.99$ , the results have the same behavior for  $K \leq 100$ . They diverge for  $K > 100$  because of the smaller  $\mu$ .

**Example 7.4.3** (continuation, Example 7.3.3)

Consider a standard 10-element linear array. We use an LCMP beamformer with derivative constraints. Assuming the signal impinges from  $u_s = 0$ ,

$$\mathbf{C} = \left[ \begin{array}{c|cc} \mathbf{1} & \dot{\mathbf{v}}_s(0) & \ddot{\mathbf{v}}_s(0) \end{array} \right] \quad (7.212)$$

and

$$\mathbf{g}^H = \left[ \begin{array}{c|c} \mathbf{1} & \mathbf{0} & \ddot{\mathbf{B}}_c(0) \end{array} \right]. \quad (7.213)$$

We use a blocking matrix consisting of the first seven columns of  $\text{orth}[\mathbf{P}_C^\perp]$ . We implement the recursion given in (7.199)–(7.206).

There are two equal-power uncorrelated interferers impinging on the array from  $u_I = 0.29, 0.45$ . We consider the following parameter values:

- (i)  $SNR = 10$  dB
- (ii)  $INR = 20$  dB
- (iii)  $\sigma_o^2/\sigma_w^2 = 10$ .

The results are shown in Figure 7.27. We see that the SMI and RLS implementations have essentially the same performance. The ad hoc fixed loading technique works well in environments where the  $SNR < INR$  and  $LNR(\sigma_L^2/\sigma_w^2)$  is chosen properly. The disadvantage is that some prior knowledge of the environment is required to choose the appropriate value of  $\sigma_L^2$ . In the next section, we revisit the quadratic constraint approach in Section 6.10 and derive a data-dependent variable loading algorithm.

### 7.4.5 Quadratically Constrained RLS

The RLS algorithm is implemented using a GSC configuration. A quadratic constraint is imposed. From (6.559)

$$\tilde{\mathbf{w}}_a = [\mathbf{S}_z + \beta \mathbf{I}]^{-1} \mathbf{p}_z, \quad (7.214)$$

and  $\beta$  is chosen so that

$$\tilde{\mathbf{w}}_a^H \tilde{\mathbf{w}}_a \leq \alpha. \quad (7.215)$$

We develop an approximate technique so that the constraint equation (6.564) does not have to be solved at each step.

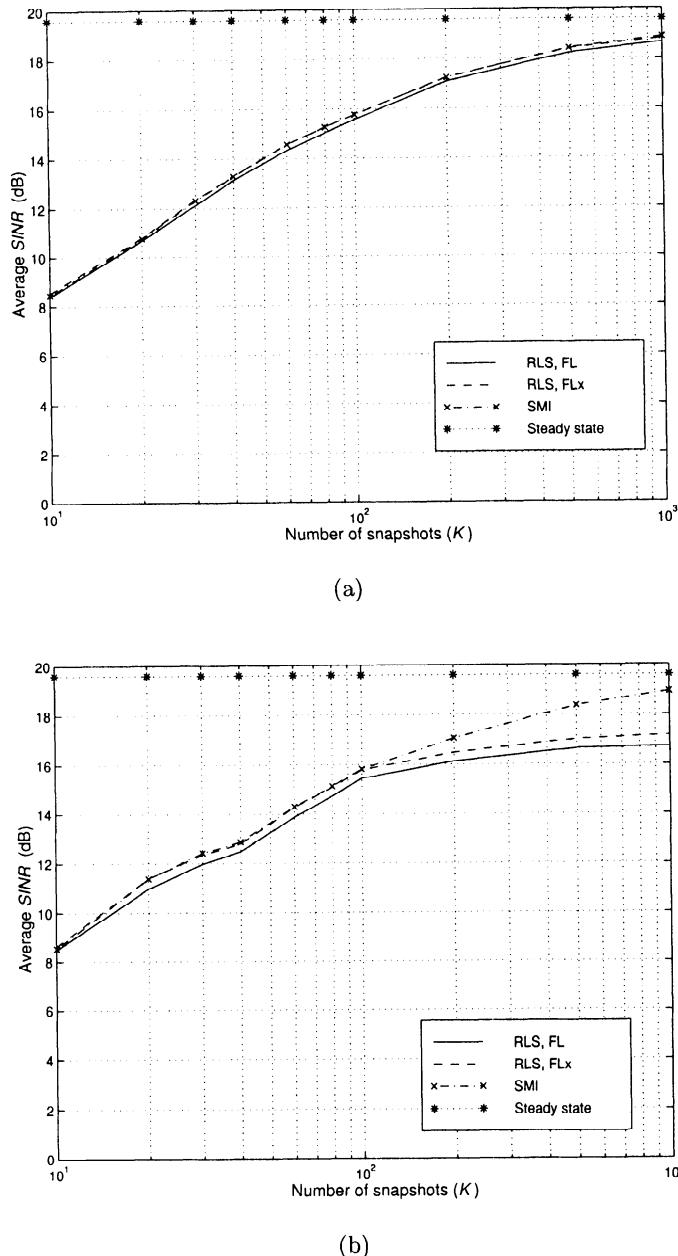


Figure 7.26 MPDR-GSC beamformer using RLS algorithm:  $SNR = 10$  dB,  $INR = 20$  dB,  $u_I = 0.29, 0.45$ ,  $\sigma_o^2/\sigma_w^2 = 10$  dB,  $\sigma_L^2/\sigma_w^2 = 10$  dB; average  $SINR_o$  versus  $K$ . (a)  $\mu = 0.999$ ; (b)  $\mu = 0.99$ .

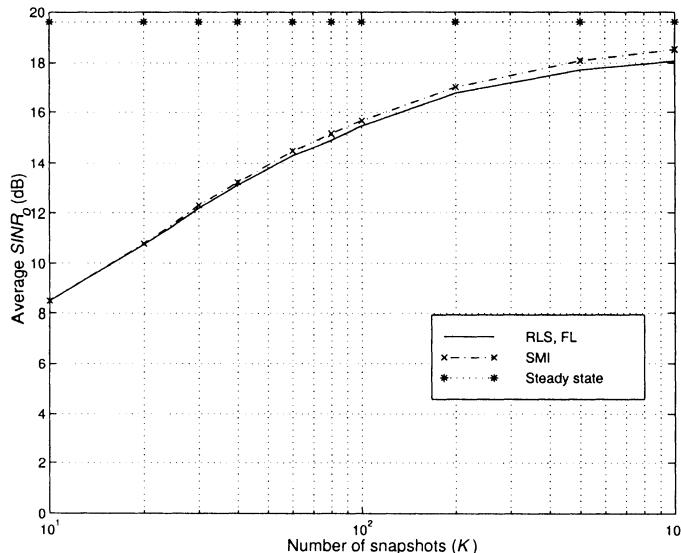


Figure 7.27 LCMP-GSC beamformer using RLS algorithm:  $SNR = 10$  dB,  $INR = 20$  dB,  $u_I = 0.29, 0.45$ ,  $\mu = 0.999$ ,  $\sigma_o^2/\sigma_w^2 = 10$  dB,  $\sigma_L^2/\sigma_w^2 = 10$  dB; average  $SINR_o$  versus  $K$ .

Rewrite (7.214) as

$$\tilde{\mathbf{w}}_a = \left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-1} \mathbf{S}_z^{-1} \mathbf{p}_z, \quad (7.216)$$

and recall that

$$\hat{\mathbf{w}}_a \triangleq \mathbf{S}_z^{-1} \mathbf{p}_z, \quad (7.217)$$

which is the optimum solution in the absence of a quadratic constraint ( $\beta = 0$ ).

Then

$$\tilde{\mathbf{w}}_a = \left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-1} \hat{\mathbf{w}}_a. \quad (7.218)$$

Now expand the  $\left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-1}$  term in a Taylor series about  $\beta = 0$ . The terms are

$$f(\beta) = \left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-1}, \quad (7.219)$$

$$f(0) = \mathbf{I}, \quad (7.220)$$

and

$$f'(\beta) = - \left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-2} \mathbf{S}_z^{-1}, \quad (7.221)$$

$$f'(0) = -\mathbf{S}_z^{-1}. \quad (7.222)$$

Retaining the first two terms,

$$\left[ \mathbf{I} + \beta \mathbf{S}_z^{-1} \right]^{-1} \simeq \mathbf{I} - \beta \mathbf{S}_z^{-1}. \quad (7.223)$$

Using (7.223) in (7.218) gives

$$\tilde{\mathbf{w}}_a \simeq \left[ \mathbf{I} - \beta \mathbf{S}_z^{-1} \right] \hat{\mathbf{w}}_a = \hat{\mathbf{w}}_a - \beta \mathbf{S}_z^{-1} \hat{\mathbf{w}}_a. \quad (7.224)$$

Defining

$$\mathbf{v} \triangleq \mathbf{S}_z^{-1} \hat{\mathbf{w}}_a, \quad (7.225)$$

(7.224) becomes

$$\tilde{\mathbf{w}}_a = \hat{\mathbf{w}}_a - \beta \mathbf{v}. \quad (7.226)$$

Using (7.226) in the constraint equation (7.215) gives

$$\tilde{\mathbf{w}}_a^H \tilde{\mathbf{w}}_a = (\hat{\mathbf{w}}_a - \beta \mathbf{v})^H (\hat{\mathbf{w}}_a - \beta \mathbf{v}) = \alpha^2, \quad (7.227)$$

which is a quadratic equation in  $\beta^2$ ,

$$\beta^2 (\mathbf{v}^H \mathbf{v}) + \beta (-2 \operatorname{Re}(\mathbf{v}^H \hat{\mathbf{w}}_a)) + (\hat{\mathbf{w}}_a^H \hat{\mathbf{w}}_a - \alpha^2) = 0. \quad (7.228)$$

Define

$$a = \mathbf{v}^H \mathbf{v}, \quad (7.229)$$

$$b = -2 \operatorname{Re}(\mathbf{v}^H \hat{\mathbf{w}}_a) = -2 \hat{\mathbf{w}}_a^H \mathbf{S}_z^{-1} \hat{\mathbf{w}}_a, \quad (7.230)$$

and

$$c = \hat{\mathbf{w}}_a^H \hat{\mathbf{w}}_a - \alpha^2. \quad (7.231)$$

Then, using the equality sign in (7.228),

$$\beta = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (7.232)$$

we observe the following characteristics:

- (i) If  $\hat{\mathbf{w}}_a^H \hat{\mathbf{w}}_a \leq \alpha^2$ , no diagonal loading is needed and we set  $\beta = 0$ . Therefore,  $c > 0$  for the cases where a non-zero  $\beta$  is needed.
- (ii)  $b \leq 0$ .
- (iii)  $a \geq 0$ .
- (iv)  $(b^2 - 4ac)$  may be positive or negative.

- (v) If  $b^2 - 4ac > 0$ , there are two real positive solutions.
- (vi) If  $b^2 - 4ac < 0$ , there are two complex conjugate solutions whose real part  $(-b/2a)$  is positive.

If condition (v) applies, choose the smallest value. This choice causes  $\tilde{\mathbf{w}}_a$  to be closest to  $\hat{\mathbf{w}}_a$ . The resulting solution meets the constraint.

If condition (vi) applies, choose the real part,

$$\beta = \frac{-b}{2a}. \quad (7.233)$$

In this case, the solution does not meet the constraint but, for a vector of the form in (7.226), it is the closest. It is convenient to write  $\beta$  as

$$\beta = \frac{-b - \operatorname{Re}(\sqrt{b^2 - 4ac})}{2a}, \quad (7.234)$$

which applies to both conditions (v) and (vi).

We refer to this algorithm as the variable loading algorithm. The next step is to apply the variable loading algorithm at each step in the RLS iteration. The basic RLS algorithm is given by (7.199)–(7.206). When the quadratic constraint is imposed, the output of (7.204) is denoted by  $\tilde{\mathbf{w}}_a(K)$  because we try to force it to satisfy the constraint in (7.215).

The steps in the variable loading algorithm can be summarized:

1. Compute

$$\tilde{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K) \left[ Y_c^*(K) - \mathbf{Z}^H(K) \hat{\mathbf{w}}_a(K-1) \right]. \quad (7.235)$$

2. Test the norm of  $\tilde{\mathbf{w}}_a(K)$ . If  $\| \tilde{\mathbf{w}}_a(K) \|^2 \leq \alpha^2$ , then

$$\hat{\mathbf{w}}_a(K) = \tilde{\mathbf{w}}_a(K). \quad (7.236)$$

3. If  $\| \tilde{\mathbf{w}}_a(K) \|^2 > \alpha^2$ , define

$$\mathbf{v}_a(K) = \mathbf{P}_z(K) \tilde{\mathbf{w}}_a(K), \quad (7.237)$$

$$a = \| \mathbf{v}_a(K) \|^2, \quad (7.238)$$

$$b = -2\operatorname{Re} \left\{ \mathbf{v}_a^H(K) \tilde{\mathbf{w}}_a(K) \right\}, \quad (7.239)$$

$$c = \| \tilde{\mathbf{w}}_a(K) \|^2 - \alpha^2, \quad (7.240)$$

$$\beta(K) = \frac{-b - \operatorname{Re} \left\{ \sqrt{b^2 - 4ac} \right\}}{2a}. \quad (7.241)$$

$$(7.242)$$

Then,

$$\hat{\mathbf{w}}_a(K) = \tilde{\mathbf{w}}_a(K) - \beta(K)\mathbf{v}_a(K). \quad (7.243)$$

Note that  $\hat{\mathbf{w}}_a(K-1)$  is used on the right side of (7.235).

A more detailed discussion on the RLS-VL algorithm is given in Tian et al. [TBV01]. A simpler technique would be to simply scale  $\tilde{\mathbf{w}}_a$ ,

$$\hat{\mathbf{w}}_a = \tilde{\mathbf{w}}_a \frac{\alpha}{\|\tilde{\mathbf{w}}_a\|}. \quad (7.244)$$

This technique was suggested by Cox et al. [CZO87] in conjunction with the LMS algorithm. Note that  $\tilde{\mathbf{w}}_a$  is in a space that is orthogonal to  $\mathbf{v}_s$ , so scaling  $\hat{\mathbf{w}}_a$  does not affect the distortionless constraint. In Section 7.7, we find it to be effective for that application. It does not appear to be effective with the RLS algorithm.

#### 7.4.6 Conjugate Symmetric Beamformers

All of the discussion to this point applied to arbitrary arrays, although we frequently used linear arrays as examples. Recall that, whenever the array is symmetric about the origin, the array manifold vectors are conjugate symmetric and  $\mathbf{S}_x$  (and  $\hat{\mathbf{S}}_x$ ) are Hermitian persymmetric. In Section 7.2, we observed that the constrained ML estimate of  $\mathbf{S}_x$  was obtained by FB averaging of the data.

$$\begin{aligned} \tilde{\mathbf{S}}_x &= \frac{1}{2} [\mathbf{C}_x + \mathbf{J}\mathbf{C}_x^*\mathbf{J}] \\ &= \frac{1}{2K} \sum_{k=1}^K \left[ \mathbf{X}(k)\mathbf{X}^H(k) + \mathbf{J}\mathbf{X}^*(k)\mathbf{X}^T(k)\mathbf{J} \right]. \end{aligned} \quad (7.245)$$

In our discussion of SMI algorithms in Section 7.3.1 (e.g., Example 7.3.2 and Figure 7.5) there was significant improvement obtained by using FB averaging. We demonstrate a similar improvement for RLS beamformers and show how to implement the algorithm.

Note that “forward-backward” was used to be consistent with the temporal literature. We are really averaging across the array in opposite directions. The corresponding weighted average is

$$\tilde{\Phi}(K) = \frac{1}{2} \sum_{k=1}^K \mu^{K-k} \left[ \mathbf{X}(k)\mathbf{X}^H(k) + \mathbf{J}\mathbf{X}^*(k)\mathbf{X}^T(k)\mathbf{J} \right]. \quad (7.246)$$

Historically, conjugate symmetry was first utilized in the parameter estimation problem (e.g., Evans et al. [EJS82]). We discuss this application in Chapters 8 and 9. The application to adaptive beamformers is due to Huarng and Yeh [HY91], and our discussion follows that reference.

Consider the direct form implementation of the MPDR beamformer using (7.245). Then,

$$\tilde{\mathbf{w}}_{mpdr} = \Lambda \tilde{\Phi}^{-1} \mathbf{v}_s, \quad (7.247)$$

and the weight vector  $\tilde{\mathbf{w}}_{mpdr}$  is conjugate symmetric.

We manipulate the data into a form where real computations can be utilized. We use the unitary transformation defined in (7.58) and (7.59) to accomplish this goal.  $\mathbf{Q}$  is defined as

$$\mathbf{Q} = \begin{cases} \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{J} & -j\mathbf{J} \end{bmatrix}, & \text{for even } N, \\ \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{0} & j\mathbf{I} \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ \mathbf{J} & \mathbf{0} & -j\mathbf{J} \end{bmatrix}, & \text{for odd } N. \end{cases} \quad (7.248)$$

Note that the  $\mathbf{I}$  and  $\mathbf{J}$  matrices have dimension  $N/2 \times N/2$ .  $\mathbf{Q}$  has two important features,

$$\mathbf{Q}^H = \mathbf{Q}^{-1}, \quad (7.249)$$

and

$$\mathbf{Q}^* = \mathbf{J}\mathbf{Q}. \quad (7.250)$$

The transformed quantities are given by

$$\bar{\mathbf{v}}_s = \mathbf{Q}^H \mathbf{v}_s, \quad (7.251)$$

$$\bar{\Phi} = \mathbf{Q}^H \tilde{\Phi} \mathbf{Q}, \quad (7.252)$$

$$\bar{\mathbf{w}} = \mathbf{Q}^H \tilde{\mathbf{w}} = \bar{\Lambda} \bar{\Phi}^{-1} \bar{\mathbf{v}}_s, \quad (7.253)$$

and

$$\bar{\Lambda} = [\bar{\mathbf{v}}_s^T \bar{\Phi}^{-1} \bar{\mathbf{v}}_s]^{-1}. \quad (7.254)$$

One can show easily that  $\bar{\mathbf{v}}_s$  is a real vector and  $\bar{\Phi}$  is a real symmetric matrix. Therefore (7.253) can be solved using real computation. The complex weight vector  $\tilde{\mathbf{w}}$  is given by

$$\tilde{\mathbf{w}} = \mathbf{Q} \bar{\mathbf{w}}. \quad (7.255)$$

The weight vector  $\tilde{\mathbf{w}}$  is used to process the input data. We now discuss how to compute  $\bar{\Phi}^{-1}$  iteratively using real computation. The approach is analogous to the iterative procedure in Section 7.3. We can write  $\bar{\Phi}(K)$  at the  $K$ th snapshot as

$$\begin{aligned}\bar{\Phi}(K) = \mu\bar{\Phi}(K-1) + \frac{1}{2}\mathbf{Q}^H & [\mathbf{X}(K)\mathbf{X}^H(K) \\ & + \mathbf{J}\mathbf{X}^*(K)\mathbf{X}^T(K)\mathbf{J}] \mathbf{Q}.\end{aligned}\quad (7.256)$$

There are various ways to update  $\bar{\Phi}^{-1}(K)$  from  $\bar{\Phi}^{-1}(K-1)$ . We first manipulate  $\bar{\Phi}(K)$  into a form that will only utilize real calculation. We rewrite the term in the brackets of (7.256) as

$$\begin{aligned}\mathbf{X}(K)\mathbf{X}^H(K) + \mathbf{J}\mathbf{X}^*(K)\mathbf{X}^T(K)\mathbf{J} = \frac{1}{2} & \left\{ [\mathbf{X}(K) + \mathbf{J}\mathbf{X}^*(K)] \right. \\ & \cdot [\mathbf{X}(K) + \mathbf{J}\mathbf{X}^*(K)]^H \\ & + [-j\mathbf{X}(K) + j\mathbf{J}\mathbf{X}^*(K)] \\ & \cdot [-j\mathbf{X}(K) + j\mathbf{J}\mathbf{X}^*(K)]^H \left. \right\},\end{aligned}\quad (7.257)$$

and define

$$\bar{\mathbf{X}}_1(K) = \frac{1}{2}\mathbf{Q}^H [\mathbf{X}(K) + \mathbf{J}\mathbf{X}^*(K)] = Re [\mathbf{Q}^H \mathbf{X}(K)],\quad (7.258)$$

and

$$\bar{\mathbf{X}}_2(K) = \frac{1}{2}\mathbf{Q}^H [-j\mathbf{X}(K) + j\mathbf{J}\mathbf{X}^*(K)] = Im [\mathbf{Q}^H \mathbf{X}(K)].\quad (7.259)$$

Now (7.256) can be written as

$$\bar{\Phi}(K) = \mu\bar{\Phi}(K-1) + [\bar{\mathbf{X}}_1(K)\bar{\mathbf{X}}_1^T(K) + \bar{\mathbf{X}}_2(K)\bar{\mathbf{X}}_2^T(K)].\quad (7.260)$$

We can now update  $\bar{\Phi}^{-1}(K)$  using any convenient recursive formula. A straightforward approach is to do two rank-one updates using the matrix inversion lemma. We can modify the recursion in (7.153)–(7.159). The first update is

$$\bar{\mathbf{P}}_1(K) = \mu^{-1}\bar{\mathbf{P}}(K-1) - \mu^{-1}\bar{\mathbf{g}}_1(K)\bar{\mathbf{X}}_1^T(K)\bar{\mathbf{P}}(K-1),\quad (7.261)$$

$$\bar{\mathbf{g}}_1(K) = \frac{\mu^{-1}\bar{\mathbf{P}}(K-1)\bar{\mathbf{X}}_1(K)}{1 + \mu^{-1}\bar{\mathbf{X}}_1^T(K)\bar{\mathbf{P}}(K-1)\bar{\mathbf{X}}_1(K)},\quad (7.262)$$

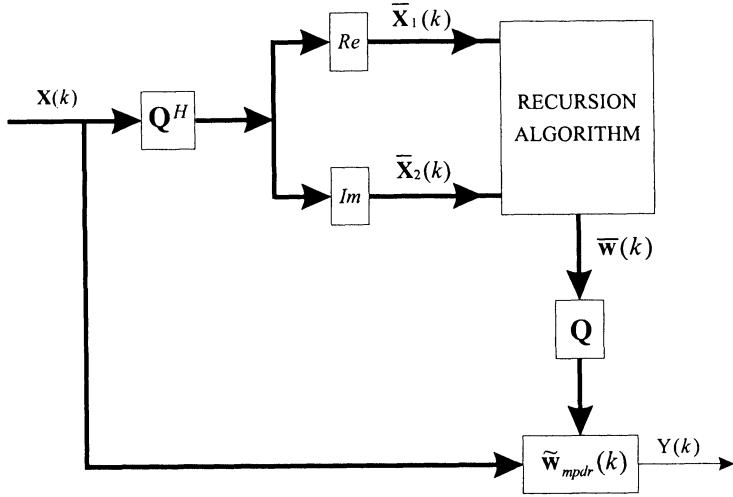


Figure 7.28 Recursive algorithm with FB averaging.

and

$$\bar{\mathbf{w}}_1(K) = \frac{\bar{\Lambda}_1(K)}{\mu\bar{\Lambda}(K-1)} [\mathbf{I} - \bar{\mathbf{g}}_1(K)\bar{\mathbf{X}}_1^T(K)] \bar{\mathbf{w}}(K-1). \quad (7.263)$$

The second update is

$$\bar{\mathbf{P}}(K) = \bar{\mathbf{P}}_1(K) - \bar{\mathbf{g}}(K)\bar{\mathbf{X}}_2^T(K)\bar{\mathbf{P}}_1(K), \quad (7.264)$$

$$\bar{\mathbf{g}}(K) = \frac{\bar{\mathbf{P}}_1(K)\bar{\mathbf{X}}_2(K)}{1 + \bar{\mathbf{X}}_2^T(K)\bar{\mathbf{P}}_1(K)\bar{\mathbf{X}}_2(K)}, \quad (7.265)$$

and

$$\bar{\mathbf{w}}(K) = \frac{\bar{\Lambda}(K)}{\bar{\Lambda}_1(K-1)} [\mathbf{I} - \bar{\mathbf{g}}(K)\bar{\mathbf{X}}_2^T(K)] \bar{\mathbf{w}}_1(K). \quad (7.266)$$

After the recursion in (7.261)–(7.266) at a particular  $K$ , we find

$$\tilde{\mathbf{w}}_{mpdr}(K) = \mathbf{Q}\bar{\mathbf{w}}(K), \quad (7.267)$$

and the output is

$$Y(K) = \tilde{\mathbf{w}}_{mpdr}^H(K)\mathbf{X}(K). \quad (7.268)$$

The beamformer is shown in Figure 7.28.

We consider a simple example to indicate the performance improvement that can be achieved.

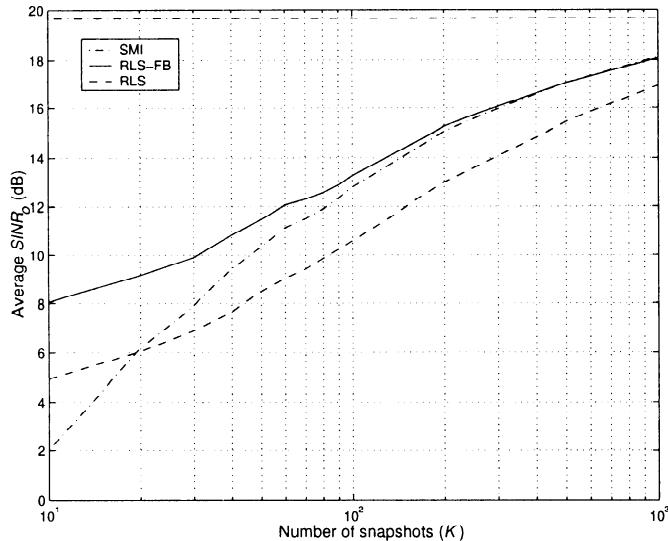


Figure 7.29 MPDR beamformer using RLS algorithm with FB averaging:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB each,  $\sigma_o^2/\sigma_w^2 = 10$  dB,  $\mu = 0.999$ , 200 trials; average  $SINR_o$  versus  $K$ .

#### Example 7.4.4 (continuation, Example 7.4.1)

Consider a standard 10-element linear array. We use the same signal and interference model as in Example 7.4.1. We implement an MPDR beamformer using the RLS algorithm with FB averaging. The results are shown in Figure 7.29.

We see that RLS-FB with  $K$  snapshots has the same performance as RLS with  $2K$  snapshots. The advantage is somewhat greater at small values of  $K$ .

#### 7.4.7 Summary

In this section, we have developed the least squares estimation version of the adaptive beamformer problem. We then developed a recursive least squares implementation of the beamformer and compared its performance to the SMI beamformers in Section 7.3. We found that by the use of appropriate diagonal loading, the RLS and SMI beamformers have comparable performance.

We developed the generalized sidelobe canceller implementation and compared the RLS implementation to the SMI implementation.

We considered the case of conjugate symmetric arrays and developed an

algorithm that utilized real arithmetic. The result provided a computational saving and improved performance.

In the next section, we develop more efficient recursive algorithms.

## 7.5 Efficient Recursive Implementation Algorithms

### 7.5.1 Introduction

In Section 7.4.2, we developed a recursive implementation of the sample matrix inversion algorithm. The implementation corresponded to a rank-one updating of  $\Phi^{-1}(K)$ . In this section, we focus on algorithms that operate on the exponentially weighted data matrix  $\mathbf{A}_\mu(K)$ , which is a  $K \times N$  complex matrix,

$$\mathbf{A}_\mu(K) \stackrel{\Delta}{=} \boldsymbol{\mu} \mathbf{A}(K), \quad (7.269)$$

where

$$\boldsymbol{\mu}(K) \stackrel{\Delta}{=} \text{diag} \left[ \begin{array}{cccc} \mu^{\frac{K-1}{2}} & \mu^{\frac{K-2}{2}} & \cdots & 1 \end{array} \right], \quad (7.270)$$

and <sup>13</sup>

$$\mathbf{A}(K) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{X}^T(1) \\ \mathbf{X}^T(2) \\ \vdots \\ \mathbf{X}^T(K) \end{bmatrix}. \quad (7.271)$$

From (7.138), we observe that

$$\Phi^*(K) = \mathbf{A}_\mu^H(K) \mathbf{A}_\mu(K). \quad (7.272)$$

The development of efficient numerically stable recursive algorithms has received significant attention over the last several decades because of their widespread application in the areas of adaptive filtering, adaptive beamforming, and system identification.

The topic is treated extensively in most textbooks on adaptive filters; e.g., Haykin ([Hay91], [Hay96]), Proakis et al. [PRLN92], Widrow and Sterns [WS85], Orfandis [Orf88], Kalouptsidis and Theodoridis [KT93], Honig and Messerschmitt [HM84], Alexander [Ale86], and Treichler et al. [TJL87].

There are numerous papers dealing with specific algorithms that we will indicate as we develop the various algorithms. Various other papers (e.g.,

---

<sup>13</sup>The  $\mathbf{A}(K)$  data matrix is related to the data matrix  $\tilde{\mathbf{X}}(K)$  defined in (7.4) by  $\mathbf{A}^T(K) = \sqrt{K} \tilde{\mathbf{X}}(K)$ . The  $\mathbf{A}(K)$  notation is more commonly used in the QR decomposition (QRD) literature.

Yuen [Yue91] and Sayed and Kailath [SK94]) show how the various algorithms are related. Godara [God74] showed the relationship to Kalman filtering.

As we discuss the various algorithms, it is useful to remember that, if we had perfect numerical precision and no perturbations in the model, then the performance of the various implementations would be the same. The issues of interest are:

- (i) Numerical stability;
- (ii) Computational complexity;
- (iii) Capability of parallel computation to improve speed and allow real-time implementation.

In Section 7.5.2, we develop a recursive algorithm that is referred to as the **QR decomposition** (QRD) or square-root algorithm. It provides an efficient implementation for both LCMP and LSE narrowband beamformers. In the text, the LSE version is developed. It can be used in the lower branch of the GSC beamformer to implement LCMP beamformers.

### 7.5.2 QR Decomposition (QRD)

In this section, we show how the QR decomposition in Section A.6 can be utilized in a recursive algorithm. Our discussion follows McWhirter and Prouder's chapter in [KT93] (see Chapter 7, pp. 260–321). We should note that the authors of that chapter developed many of the original results in the application of QRD techniques to adaptive beamforming and the implementation of those techniques.

We formulate the problem as a least squares estimation problem (see Section 7.4.1). Using the transpose of (7.134), we can write the output of the beamformer at time  $k$  as<sup>14</sup>

$$Y(k) = \mathbf{X}^T(k)\mathbf{w}^*(K), \quad (7.273)$$

and, from (7.147), the error is

$$e(k) = D(k) - \mathbf{X}^T(k)\mathbf{w}^*(K), \quad k = 1, \dots, K. \quad (7.274)$$

As in (7.148), we minimize a weighted residual error

$$\xi_\mu(K) = \|\mathbf{e}_\mu(K)\|^2, \quad (7.275)$$

---

<sup>14</sup>We use this form rather than  $\mathbf{w}^H\mathbf{X}(K)$  for convenience in presenting the final processor.

where

$$\mathbf{e}_\mu(K) = \boldsymbol{\mu}(K) \begin{bmatrix} e(1) & e(2) & \cdots & e(K) \end{bmatrix}^T. \quad (7.276)$$

The error vector in (7.276) can be written as

$$\mathbf{e}_\mu(K) = \mathbf{d}_\mu(K) - \mathbf{A}_\mu(K)\mathbf{w}^*(K), \quad (7.277)$$

where

$$\mathbf{A}_\mu(K) = \boldsymbol{\mu}(K) \begin{bmatrix} \mathbf{X}^T(1) \\ \mathbf{X}^T(2) \\ \vdots \\ \mathbf{X}^T(K) \end{bmatrix}, \quad (7.278)$$

and

$$\mathbf{d}_\mu(K) = \boldsymbol{\mu}(K) \begin{bmatrix} D(1) \\ D(2) \\ \vdots \\ D(K) \end{bmatrix}. \quad (7.279)$$

The subscript “ $\mu$ ” denotes the inclusion of the exponential weighting.

The solution to the minimization problem was given in (7.184), which can be rewritten as

$$\boldsymbol{\Phi}^*(K)\hat{\mathbf{w}}_{lse}^*(K) = \boldsymbol{\Phi}_{\mathbf{x}d^*}^*(K). \quad (7.280)$$

Using (7.272) and (7.4) in (7.280) gives

$$\mathbf{A}_\mu^H(K)\mathbf{A}_\mu(K)\hat{\mathbf{w}}_{lse}^*(K) = \mathbf{A}_\mu^H(K)\mathbf{d}_\mu(K). \quad (7.281)$$

We now develop an alternative approach to the recursion algorithm in Section 7.4 based on the QRD, which has good numerical properties. As discussed in Section A.6, we can find a  $K \times K$  unitary matrix,  $\mathbf{Q}(K)$  such that

$$\mathbf{Q}(K)\mathbf{A}_\mu(K) = \begin{bmatrix} \tilde{\mathbf{R}}(K) \\ \mathbf{0} \end{bmatrix}, \quad (7.282)$$

where  $\tilde{\mathbf{R}}(K)$  is an  $N \times N$  upper triangular matrix and  $\mathbf{0}$  is a  $(K - N) \times N$  matrix of zeros. Because  $\mathbf{Q}(K)$  is unitary,  $\tilde{\mathbf{R}}(K)$  is just the Cholesky factor of the data covariance matrix  $\boldsymbol{\Phi}(K)$ . Using (7.249), (7.281) can be written as

$$\mathbf{A}_\mu^H(K)\mathbf{Q}^H\mathbf{Q}\mathbf{A}_\mu(K)\hat{\mathbf{w}}_{lse}^*(K) = \mathbf{A}_\mu^H(K)\mathbf{Q}^H\mathbf{Q}\mathbf{d}_\mu(K). \quad (7.283)$$

Substituting (7.282) into (7.283) gives

$$\begin{bmatrix} \tilde{\mathbf{R}}(K) \\ \mathbf{0} \end{bmatrix}^H \begin{bmatrix} \tilde{\mathbf{R}}(K) \\ \mathbf{0} \end{bmatrix} \hat{\mathbf{w}}_{lse}^*(K) = \begin{bmatrix} \tilde{\mathbf{R}}(K) \\ \mathbf{0} \end{bmatrix}^H \mathbf{Q}\mathbf{d}_\mu(K). \quad (7.284)$$

The  $K \times 1$  vector,  $\mathbf{Qd}_\mu(K)$  can be partitioned as

$$\mathbf{Qd}_\mu(K) = \begin{bmatrix} \mathbf{p}(K) \\ \mathbf{v}(K) \end{bmatrix}, \quad (7.285)$$

where  $\mathbf{p}(K)$  is an  $N \times 1$  vector. Then, (7.284) reduces to

$$\tilde{\mathbf{R}}(K)\hat{\mathbf{w}}_{lse}^*(K) = \mathbf{p}(K), \quad (7.286)$$

which is straightforward to solve because of the upper triangular structure of  $\tilde{\mathbf{R}}(K)$ . The minimum weighted error can be written as

$$\xi_o(K) = \|\mathbf{v}(K)\|^2. \quad (7.287)$$

In order to use (7.286), the QRD in (7.282) must be implemented.

As discussed in Section A.6, the triangularization can be accomplished with either a Givens rotation (see Section A.6.2) or a Householder transformation (see Section A.6.3). We shall find that the Givens rotation is particularly suited to adaptive beamforming because it leads to an efficient algorithm for recursive updating.

To develop the recursive algorithm, we assume that  $\mathbf{A}_\mu(K-1)$  has already been reduced to a triangular form by the unitary transformation,

$$\mathbf{Q}(K-1)\mathbf{A}_\mu(K-1) = \begin{bmatrix} \tilde{\mathbf{R}}(K-1) \\ \mathbf{0} \end{bmatrix}. \quad (7.288)$$

First, define a  $K \times K$  unitary matrix,

$$\bar{\mathbf{Q}}(K-1) = \begin{bmatrix} \mathbf{Q}(K-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (7.289)$$

and write  $\mathbf{A}_\mu(K)$  as,

$$\mathbf{A}_\mu(K) = \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{A}_\mu(K-1) \\ \mathbf{X}^T(K) \end{bmatrix}. \quad (7.290)$$

Then,

$$\begin{aligned} \bar{\mathbf{Q}}(K-1)\mathbf{A}_\mu(K) &= \bar{\mathbf{Q}}(K-1) \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{A}_\mu(K-1) \\ \mathbf{X}^T(K) \end{bmatrix} \\ &= \begin{bmatrix} \mu^{\frac{1}{2}}\tilde{\mathbf{R}}(K-1) \\ \mathbf{0} \\ \mathbf{X}^T(K) \end{bmatrix}. \end{aligned} \quad (7.291)$$

The required triangularization can be completed by using a sequence of complex Givens rotations to eliminate  $\mathbf{X}^T(K)$ . We demonstrate the procedure with a simple example.

**Example 7.5.1**

We assume  $N = 3$  and  $K = 5$ . Then, the right side of (7.291) can be written as

$$\mathbf{M}_1 = \begin{bmatrix} \mu^{\frac{1}{2}}\tilde{r}_{11} & \mu^{\frac{1}{2}}\tilde{r}_{12} & \mu^{\frac{1}{2}}\tilde{r}_{13} \\ 0 & \mu^{\frac{1}{2}}\tilde{r}_{22} & \mu^{\frac{1}{2}}\tilde{r}_{23} \\ 0 & 0 & \mu^{\frac{1}{2}}\tilde{r}_{33} \\ 0 & 0 & 0 \\ x_1(5) & x_2(5) & x_3(5) \end{bmatrix}. \quad (7.292)$$

The  $\tilde{r}_{ii}$  are real and the  $\tilde{r}_{ij}$  ( $i \neq j$ ) and the  $x_i$  are complex. The first Givens rotation uses the first and last rows to eliminate  $x_1(5)$ . Thus,

$$\mathbf{G}_1 = \begin{bmatrix} c_1 & \mathbf{0}^T & s_1^* \\ & 1 & \\ \mathbf{0} & 1 & \mathbf{0} \\ -s_1 & \mathbf{0}^T & c_1 \end{bmatrix}, \quad (7.293)$$

where

$$|c_1|^2 + |s_1|^2 = 1, \quad (7.294)$$

and  $c_1$  can be real without loss of generality.

$$c_1 = \frac{\mu^{\frac{1}{2}}\tilde{r}_{11}}{\sqrt{\mu\tilde{r}_{11}^2 + |x_1|^2}}, \quad (7.295)$$

and

$$s_1 = \frac{x_1}{\sqrt{\mu\tilde{r}_{11}^2 + |x_1|^2}}, \quad (7.296)$$

where we have suppressed the  $K = 5$  argument. Multiplying  $\mathbf{M}_1$  by  $\mathbf{G}_1$  and using (7.295) and (7.296) gives

$$\begin{aligned} \mathbf{G}_1 \mathbf{M}_1 &= \begin{bmatrix} c_1\mu^{\frac{1}{2}}\tilde{r}_{11} + s_1^*x_1 & c_1\mu^{\frac{1}{2}}\tilde{r}_{12} + s_1^*x_2 & c_1\mu^{\frac{1}{2}}\tilde{r}_{13} + s_1^*x_3 \\ 0 & \mu^{\frac{1}{2}}\tilde{r}_{22} & \mu^{\frac{1}{2}}\tilde{r}_{23} \\ 0 & 0 & \mu^{\frac{1}{2}}\tilde{r}_{33} \\ 0 & 0 & 0 \\ \tilde{r}'_{11} & \tilde{r}'_{12} & \tilde{r}'_{13} \\ 0 & \mu^{\frac{1}{2}}\tilde{r}_{22} & \mu^{\frac{1}{2}}\tilde{r}_{23} \\ 0 & 0 & \mu^{\frac{1}{2}}\tilde{r}_{33} \\ 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{r}'_{11} & \tilde{r}'_{12} & \tilde{r}'_{13} \\ 0 & \mu^{\frac{1}{2}}\tilde{r}_{22} & \mu^{\frac{1}{2}}\tilde{r}_{23} \\ 0 & 0 & \mu^{\frac{1}{2}}\tilde{r}_{33} \\ 0 & 0 & 0 \\ 0 & x'_2 & x'_3 \end{bmatrix}. \end{aligned} \quad (7.297)$$

Note that the first element in the bottom row is eliminated and the elements in both the first row and the last row are changed.

The second Givens rotation uses the second row and the last row to eliminate  $x'_2$ :

$$\mathbf{G}_2 = \begin{bmatrix} 1 & & \mathbf{0}^T \\ & \cdots & \\ \mathbf{0} & c_2 & 0 & 0 & s_2^* \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \\ & -s_2 & 0 & 0 & c_2 \end{bmatrix}. \quad (7.298)$$

Choosing  $c_2$  and  $s_2$  in a manner analogous to (7.295) and (7.296), we obtain

$$\mathbf{G}_2 \mathbf{G}_1 \mathbf{M}_1 = \begin{bmatrix} \tilde{r}'_{11} & \tilde{r}'_{12} & \tilde{r}'_{13} \\ 0 & \tilde{r}'_{22} & \tilde{r}'_{23} \\ 0 & 0 & \mu^{\frac{1}{2}} \tilde{r}_{33} \\ 0 & 0 & 0 \\ 0 & 0 & x''_3 \end{bmatrix}. \quad (7.299)$$

Note that the first row is unchanged and the second and bottom rows are changed. Similarly,

$$\mathbf{G}_3 = \begin{bmatrix} 1 & \mathbf{0} & & \\ & 1 & & \\ \mathbf{0} & c_3 & 0 & s_3^* \\ & 0 & 1 & 0 \\ & -s_3 & 0 & c_3 \end{bmatrix}, \quad (7.300)$$

and

$$\mathbf{G}_3 \mathbf{G}_2 \mathbf{G}_1 \mathbf{M}_1 = \begin{bmatrix} \tilde{\mathbf{R}}(N) \\ \mathbf{0} \end{bmatrix}, \quad (7.301)$$

which is the desired result. Note that the  $\mathbf{G}_3$  operation only changes the third row and the bottom row.

In general, we can write

$$\mathbf{Q}(K) = \hat{\mathbf{Q}}(K) \bar{\mathbf{Q}}(K-1), \quad (7.302)$$

where

$$\hat{\mathbf{Q}}(K) = \mathbf{G}_N \cdots \mathbf{G}_2 \mathbf{G}_1, \quad (7.303)$$

and

$$\hat{\mathbf{Q}}(K) \begin{bmatrix} \mu^{\frac{1}{2}} \tilde{\mathbf{R}}(K-1) \\ \mathbf{0} \\ \mathbf{X}^T(K) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}(K) \\ \mathbf{0} \\ \mathbf{0}^T \end{bmatrix}, \quad (7.304)$$

which is the desired result. Note that each Givens rotation only changes the row corresponding to its subscript and the bottom row.

The term in (7.285) must also be computed recursively. The same  $\mathbf{Q}(K)$  can be used to perform the recursion on  $\mathbf{p}(K)$ . At the  $(K - 1)$  iteration,

$$\mathbf{Q}(K - 1)\mathbf{d}_\mu(K - 1) = \begin{bmatrix} \mathbf{p}(K - 1) \\ \mathbf{v}(K - 1) \end{bmatrix}, \quad (7.305)$$

where  $\mathbf{p}(K - 1)$  is  $N \times 1$  and  $\mathbf{v}(K - 1)$  is  $(K - 1 - N) \times 1$ . The matrix  $\bar{\mathbf{Q}}(K - 1)$  is defined in (7.289). Write

$$\bar{\mathbf{Q}}(K - 1)\mathbf{d}_\mu(K - 1) = \bar{\mathbf{Q}}(K - 1) \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{d}_\mu(K - 1) \\ D(K) \end{bmatrix}. \quad (7.306)$$

Substituting (7.289) into (7.306) gives

$$\begin{aligned} \mathbf{Q}(K - 1)\mathbf{d}_\mu(K - 1) &= \begin{bmatrix} \mathbf{Q}(K - 1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{d}_\mu(K - 1) \\ D(K) \end{bmatrix} \\ &= \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{p}(K - 1) \\ \mu^{\frac{1}{2}}\mathbf{v}(K - 1) \\ D(K) \end{bmatrix}. \end{aligned} \quad (7.307)$$

Now apply  $\hat{\mathbf{Q}}(K)$  from (7.303),

$$\hat{\mathbf{Q}}(K) \begin{bmatrix} \mu^{\frac{1}{2}}\mathbf{p}(K - 1) \\ \mu^{\frac{1}{2}}\mathbf{v}(K - 1) \\ D(K) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(K) \\ \mu^{\frac{1}{2}}\mathbf{v}(K - 1) \\ \tilde{e}(K) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(K) \\ \mathbf{v}(K) \\ \tilde{e}(K) \end{bmatrix}. \quad (7.308)$$

Multiplication by  $\mathbf{G}_1$  updates  $[\mathbf{p}(K)]_1$  and changes the last element in the overall vector. Multiplication by  $\mathbf{G}_2$  updates  $[\mathbf{p}(K)]_2$  and changes the last element in the overall vector. Continuing, multiplication by  $\mathbf{G}_N$  updates  $[\mathbf{p}(K)]_N$  and generates  $\tilde{e}(K)$  (whose significance is discussed shortly). The other elements of  $\mathbf{v}(K)$  do not need to be computed.

This discussion shows that by applying  $\mathbf{Q}(K)$  recursively to  $A_\mu(K)$  and  $\mathbf{d}_\mu(K)$ , the  $\tilde{\mathbf{R}}(K)$  and  $\mathbf{p}(K)$  required to solve (7.286) can be generated. However, actually using  $\mathbf{Q}(K)$  is inefficient. First, observe that the right side of (7.291) is already available from the  $(K - 1)$  step, so the  $\bar{\mathbf{Q}}(K - 1)$  multiplication is not necessary. Now, examine the Givens rotation in Example 7.5.1:

- (i) The  $\mathbf{G}_1$  operation consists of computing  $c_1$  and  $s_1$  and then computing three new elements in the top row and two new elements in the bottom row. The left element in the bottom row becomes zero, and all other elements are unchanged.

- (ii) The  $\mathbf{G}_2$  operation consists of computing  $c_2$  and  $s_2$  and then computing two new elements in the second row and one new element in the bottom row.
- (iii) The  $\mathbf{G}_3$  operation consists of computing  $c_3$  and  $s_3$  and then computing one new element in the third row.

The formulas for each of these calculations are given in Section A.6.3. These formulas are implemented instead of the complete matrix multiplication. A similar discussion applies to  $\mathbf{p}(K)$  updates.

At this point,  $\hat{\mathbf{w}}_{lse}(K)$  can be obtained by solving (7.286). The solution is straightforward because of the upper triangular structure of  $\tilde{\mathbf{R}}(K)$ . However,  $\hat{d}(K)$  can be generated without finding an explicit expression for  $\hat{\mathbf{w}}_{lse}(K)$ .

The error at time  $K$  is

$$\mathbf{e}(K) = \mathbf{D}(K) - \mathbf{X}^T(K)\hat{\mathbf{w}}_{lse}^*(K). \quad (7.309)$$

Write  $\hat{\mathbf{Q}}(K)$  as a partitioned  $K \times K$  matrix

$$\hat{\mathbf{Q}}(K) = \begin{bmatrix} \Sigma(K) & \mathbf{0}^T & \mathbf{q}(K) \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \sigma^T(K) & \mathbf{0}^T & \tilde{\alpha}(K) \end{bmatrix}, \quad (7.310)$$

where  $\Sigma(K)$  is  $N \times N$ ,  $\mathbf{q}(K)$  and  $\sigma(K)$  are  $N \times 1$ , and  $\tilde{\alpha}(K)$  is a scalar. The first step is to find  $\tilde{\alpha}(K)$ .

From (7.303),

$$\hat{\mathbf{Q}}(K) = \prod_{i=1}^N \mathbf{G}_i, \quad (7.311)$$

where the  $\mathbf{G}_i$  are the Givens rotation matrices. They have two elements on the diagonal equal to  $c_i$ , and the remaining elements are one. They have only two non-zero off-diagonal elements  $s_i^*$  and  $-s_i$ . The model from Example 7.5.1 illustrates their structure.

**Example 7.5.2** (continuation)

$$\hat{\mathbf{Q}}(5) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & s_3^* \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -s_3 & 0 & c_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & s_2^* \\ \mathbf{0} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -s_2 & 0 & 0 & c_2 \end{bmatrix} \begin{bmatrix} c_1 & 0 & 0 & 0 & s_1^* \\ 0 & \mathbf{I} & & & \mathbf{0} \\ -s_1 & 0 & 0 & 0 & c_1 \end{bmatrix}. \quad (7.312)$$

Thus,

$$\tilde{\alpha}(5) = c_1 c_2 c_3. \quad (7.313)$$

In general,

$$\tilde{\alpha}(K) = \prod_{i=1}^N c_i, \quad (7.314)$$

where the  $c_i$  are the cosine factors in the Givens rotations.

Next, multiply both sides of (7.304) by  $\hat{\mathbf{Q}}^H(K)$  and use the unitary character of  $\hat{\mathbf{Q}}(K)$  to obtain

$$\mathbf{X}^T(K) = \mathbf{q}^H(K) \tilde{\mathbf{R}}(K). \quad (7.315)$$

Repeating the process with (7.308) gives

$$D(K) = \mathbf{q}^H(K) \mathbf{p}(K) + \tilde{\alpha}(K) \tilde{e}(K). \quad (7.316)$$

Substituting (7.315) and (7.316) into (7.309) gives

$$e(K) = -\mathbf{q}^H(K) \tilde{\mathbf{R}}(K) \mathbf{w}^*(K) + \mathbf{q}^H(K) \mathbf{p}(K) + \tilde{\alpha}(K) \tilde{e}(K). \quad (7.317)$$

However, (7.286) implies that the first two terms on the right side of (7.317) sum to zero. Thus,

$$e(K) = \tilde{\alpha}(K) \tilde{e}(K), \quad (7.318)$$

and

$$\hat{D}(K) = D(K) - e(K). \quad (7.319)$$

The significance of the result in (7.318) is that the quantities  $\tilde{\alpha}(K)$  and  $\tilde{e}(K)$  are generated by the recursive algorithm and we do not need to find  $\hat{\mathbf{w}}_{lse}(K)$ .

The last step in the development is to show how the Givens rotation algorithm may be implemented in parallel with a triangular processor array.

The triangular processor array is shown in Figure 7.30, for the case of  $N = 4$ .

The input snapshots  $\mathbf{X}^T(K)$  and the desired signal  $D(K)$  enter the array from the top. Each of the internal cells is performing a part of Givens rotation as shown in Figure 7.31 and Table 7.2.

At time  $(K - 1)$ , the left boundary cells and the internal cells have the elements of  $\mathbf{R}(K - 1)$  stored. The right boundary cells have the elements of  $\mathbf{p}(K)$  stored.

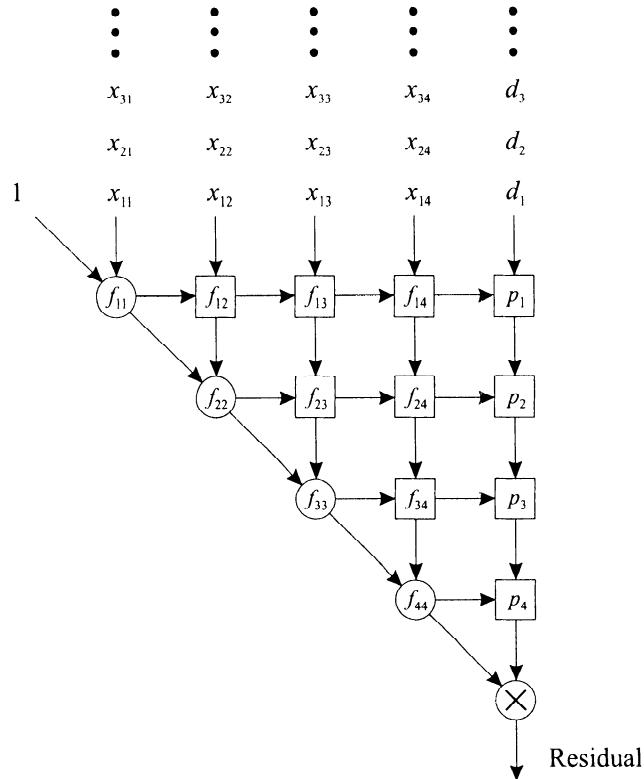


Figure 7.30 Triangular processor array.

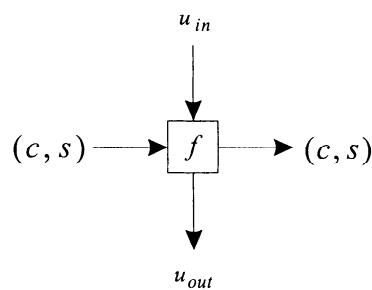


Figure 7.31 Processing elements for triangular QRD array: internal cells and right boundary cells.

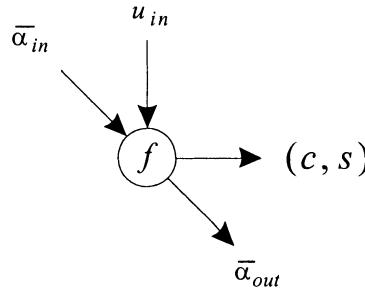


Figure 7.32 Left boundary cells.

**Table 7.2** Functions of Internal and Right Boundary Cells

|   |
|---|
| $u_{out} = -s_1 \mu^{\frac{1}{2}} \tilde{r} + c_1 u_{in}$<br>$\tilde{r}_{new} = c_1 \mu^{\frac{1}{2}} \tilde{r} + s_1^* u_{in}$ |
|---|

At time  $K$ , the new snapshot vector  $\mathbf{X}^T(K)$  and  $D(K)$  are sent to the cells in the first row. The first row of cells is performing the  $\mathbf{G}_1\mathbf{M}_1$  rotation of (7.297) (as adapted for  $N = 4$ ). Note that each cell in the row is using the same  $c_1$  and  $s_1$  values. These values are computed by the left boundary cell using (7.295) and (7.296) (we will discuss the left boundary cell in the next paragraph). As indicated in (7.297), the data output of the left boundary cell is zero. The second row of cells carries out the  $\mathbf{G}_2$  rotation in (7.298). The third and fourth rows complete the rotation. The right boundary cell is using the same rotation with an input of  $D(K)$ .

The functions in the left boundary cell are described in Figure 7.32 and Table 7.3.

The output of the  $E$  cell is  $\tilde{e}(K)$  from (7.308). Then  $\hat{D}(K)$  is obtained from (7.319).

We initialize the matrix  $\tilde{\mathbf{R}}(0)$  equal to zero. Therefore all of the  $\tilde{r}$  in the boundary cells are real and we only perform real arithmetic.

We can improve on the efficiency of the Givens QRD algorithm by devising a version that does not require the square-root operation. Gentleman [Gen73] and Hammarling [Ham74] have developed efficient algorithms using a modified Givens rotation that does not require square-root operations. The algorithm is discussed in Section 7.2.4 of McWhirter and Prouder's chapter in [KT93] and in Section 7.3.2 of [PRLN92]. The reader is referred to these

references for a discussion of the square-root-free Givens algorithm.

**Table 7.3** Description of Element Functions in Figure 7.32

| If $u_{in} = 0$                                       | If $u_{in} \neq 0$  |
|---|---|
| $c = 1$   | $c = \frac{\mu^{\frac{1}{2}} \tilde{r}_{old}}{\sqrt{\mu \tilde{r}_{old}^2 +  u_{in} ^2}}$ |
| $s = 0$   | $s = \frac{x_1}{\sqrt{\mu \tilde{r}_{old}^2 +  u_{in} ^2}}$                               |
| $\tilde{\alpha}_{out} = \tilde{\alpha}_{in}$          | $\tilde{\alpha}_{out} = c \tilde{\alpha}_{in}$  |
| $\tilde{r}_{new} = \mu^{\frac{1}{2}} \tilde{r}_{old}$ | $\tilde{r}_{new} = \sqrt{\mu \tilde{r}_{old}^2 +  u_{in} ^2}$                             |

In this section, we have developed the QRD implementation of the least squares estimation problem. It can be used in the lower branch of a GSC beamformer. A direct form LCMP or LCMV beamformer can also be implemented using a QRD. The MVDR version is derived in McWhirter and Shepherd [MS89] (see pp. 619–621 of Haykin [Hay96] for a discussion of MVDR beamformer).

The QRD technique is important because it is numerically stable and can be implemented in triangular processor arrays. A disadvantage is that it is not clear how to efficiently incorporate diagonal loading.

## 7.6 Gradient Algorithms

### 7.6.1 Introduction

In Sections 7.6 and 7.7, we discuss a group of algorithms that rely on the quadratic characteristic of the error surface and utilize gradient techniques to find the optimum weight vector. In Section 7.6, we assume that the ensemble statistics are known. We can use deterministic gradient techniques to find the optimum weight vector. In Section 7.7, we consider stochastic gradient algorithms, in which the statistics must be estimated from the data.

The two deterministic gradient techniques that are most widely used are the Newton algorithm (and various quasi-Newton algorithms) and the steepest descent algorithm. These techniques are discussed in a number of

texts dealing with classical optimization theory, as well as books on adaptive filtering (e.g., Widrow and Stearns [WS85] and Haykin [Hay96]).

We will discuss the Newton algorithm in Section 8.6 in the context of parameter estimation. In Section 7.6, we only consider the steepest descent method. Our motivation is that the stochastic version of the steepest descent algorithm is the **least-mean-square** (LMS) algorithm, which is widely used in practice.

The major advantage of steepest descent and LMS algorithms is their computational simplicity, which is  $O(N)$  in the narrowband model, where  $N$  is the number of sensors. For broadband processors, the complexity is  $O(NM)$ , where  $M$  is the number of discrete frequencies or taps that are utilized. Their major disadvantage is that their rate of convergence depends on the eigenvalue spread in  $\mathbf{S}_x$  and may be slow in a multiple interference environment.

The LMS algorithms are due to Widrow et al. [WMGG67]. Widrow references earlier papers by Shor [Sho66] and Mermoz [Mer65]. In addition to adaptive antennas, the LMS algorithm has been applied in a number of other areas including adaptive equalization, adaptive deconvolution, adaptive noise cancelling, and adaptive line enhancement (e.g., Widrow and Stearns [WS85] and Haykin [Hay96]).

At the same time, Howells and Applebaum were developing narrowband adaptive antennas that were equivalent to the LMS narrowband arrays, but their work was not publicized to the same extent (e.g., [App66], [App76], and [How76]).

In Section 7.6.2, we develop deterministic gradient algorithms for the MMSE beamformer using the method of steepest descent. In Section 7.6.3, we develop the steepest descent version of the LCMP beamformer. In Section 7.6.4, we summarize our results.

In Section 7.7, we develop various versions of the LMS algorithm, which is a stochastic gradient algorithm. For notational simplicity, Sections 7.6 to 7.7 consider narrowband arrays. In Section 7.11, we show how the LMS algorithm is applied to broadband arrays.

Although we do not discuss it in detail, we should note that if we utilized a Newton-type algorithm for the deterministic gradient algorithm, the stochastic version is called the LMS-Newton algorithm (e.g., Glentis et al. [GBT99] or Diniz et al. [DCA95]). The most common version of this algorithm is identical to the RLS algorithm in Section 7.4.

### 7.6.2 Steepest Descent: MMSE Beamformers

The simplest way to motivate the stochastic gradient algorithms that we utilize in the next section is to first consider deterministic gradient algorithms.

These deterministic gradient algorithms are discussed in most books or tutorial articles on adaptive filtering or beamforming (e.g., Widrow and Stearns [WS85], Chapter 8 of [Hay96], Monzingo and Miller [MM80], Compton [Com88], Proakis et al. [PRLN92]).

We consider two algorithms. In Section 7.6.2, we consider the MMSE beamformer and develop a steepest descent algorithm to find  $\hat{\mathbf{w}}_o$ . These results apply to LCMP beamformers if we use the generalized sidelobe canceller implementation of Figure 6.46 and can be adapted easily to accommodate quadratic constraints. In Section 7.6.3, we consider a direct form implementation of the LCMP beamformer and derive a steepest descent algorithm to find  $\hat{\mathbf{w}}_{lcmp}$ .

From Section 6.2, the MSE is<sup>15</sup>

$$\begin{aligned}\xi(\mathbf{w}) &= E[(D - \mathbf{w}^H \mathbf{X})(D^* - \mathbf{X}^H \mathbf{w})] \\ &= \sigma_d^2 - \mathbf{w}^H \mathbf{p} - \mathbf{p}^H \mathbf{w} + \mathbf{w}^H \mathbf{S}_x \mathbf{w},\end{aligned}\quad (7.320)$$

where

$$\mathbf{p} \stackrel{\Delta}{=} E[\mathbf{X} D^*] = \mathbf{S}_{xd^*}. \quad (7.321)$$

In the discussion in this section, we assume  $\mathbf{p}$  is known.

The gradient with respect to  $\mathbf{w}^H$  is

$$\nabla \xi_{\mathbf{w}^H} = -\mathbf{p} + \mathbf{S}_x \mathbf{w}. \quad (7.322)$$

Setting the gradient equal to zero gives the familiar equation for the Wiener-Hopf beamformer,

$$\mathbf{S}_x \mathbf{w}_o = \mathbf{p}. \quad (7.323)$$

The resulting MMSE is

$$\xi_o = \sigma_d^2 - \mathbf{w}_o^H \mathbf{S}_x \mathbf{w}_o = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o. \quad (7.324)$$

Instead of solving (7.323) directly by inverting  $\mathbf{S}_x$ , we solve it by a gradient search technique. The error surface is quadratic so the search procedure should converge to the unique minimum  $\xi$  point.

We should observe that there are other search techniques, such as the Newton or quasi-Newton algorithm for minimizing  $\xi(\mathbf{w})$ . We use the steepest

---

<sup>15</sup>Our discussion is similar to that in [WS85] and Haykin [Hay96].

descent technique because of its computational simplicity. The cost will be slow convergence.

We define

$$\mathbf{w}(K) = \mathbf{w}(K-1) + \alpha (-\nabla \xi_{\mathbf{w}^H}), \quad (7.325)$$

where  $\alpha$  is a real parameter, which we refer to as the step size parameter. In most cases, we use a variable step size parameter  $\alpha(K)$ . Using (7.322) in (7.325) gives

$$\mathbf{w}(K) = \mathbf{w}(K-1) + \alpha [\mathbf{p} - \mathbf{S}_x \mathbf{w}(K-1)], \quad K = 1, 2, \dots \quad (7.326)$$

In order to examine its behavior, it is convenient to define the weight-error vector,  $\mathbf{w}_e(K)$ ,

$$\mathbf{w}_e(K) \stackrel{\Delta}{=} \mathbf{w}(K) - \mathbf{w}_o. \quad (7.327)$$

Using (7.323) and (7.327) in (7.326) gives

$$\mathbf{w}_e(K) = (\mathbf{I} - \alpha \mathbf{S}_x) \mathbf{w}_e(K-1). \quad (7.328)$$

The next step is to rotate (7.328) into an orthonormal coordinate system. Using (5.237), we write

$$\mathbf{S}_x = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^H. \quad (7.329)$$

Substituting (7.329) in (7.328), multiplying by  $\mathbf{U}^H$ , and utilizing the unitary nature of  $\mathbf{U}$ , gives

$$\mathbf{U}^H \mathbf{w}_e(K) = (\mathbf{I} - \alpha \boldsymbol{\Lambda}) \mathbf{U}^H \mathbf{w}_e(K-1). \quad (7.330)$$

We now define<sup>16</sup>

$$\mathbf{v}(K) \stackrel{\Delta}{=} \mathbf{U}^H \mathbf{w}_e(K). \quad (7.331)$$

Using (7.331) in (7.330) gives

$$\mathbf{v}(K) = (\mathbf{I} - \alpha \boldsymbol{\Lambda}) \mathbf{v}(K-1). \quad (7.332)$$

Assuming  $\mathbf{w}(0) = \mathbf{0}$ , then

$$\mathbf{v}(0) = -\mathbf{U}^H \mathbf{w}_o. \quad (7.333)$$

The coefficient of  $\mathbf{v}(K-1)$  in (7.332) is diagonal so the components of  $\mathbf{v}(K)$  can be treated independently,

$$v_n(K) = (1 - \alpha \lambda_n) v_n(K-1), \quad n = 1, 2, \dots, N. \quad (7.334)$$

---

<sup>16</sup>Note that  $\mathbf{v}(K)$  is not related to the array manifold vector.

The solution to (7.334) is

$$v_n(K) = (1 - \alpha\lambda_n)^K v_n(0), \quad n = 1, 2, \dots, N. \quad (7.335)$$

All of the eigenvalues are real and positive, so  $v_n(K)$  is a geometric series. In order for the sequence to converge, we require

$$|1 - \alpha\lambda_n| < 1, \quad n = 1, 2, \dots, N, \quad (7.336)$$

which implies

$$0 < \alpha < \frac{2}{\lambda_{max}}, \quad (7.337)$$

where  $\lambda_{max}$  is the maximum eigenvalue of  $\mathbf{S}_x$ . The condition in (7.337) is necessary and sufficient.

We can fit a continuous envelope to the geometric series (e.g., p. 347 of [Hay96]),

$$1 - \alpha\lambda_n = \exp\left(-\frac{1}{\tau_n}\right), \quad (7.338)$$

where  $\tau_n$  is the time constant,

$$\tau_n = \frac{-1}{\ln(1 - \alpha\lambda_n)} \quad (7.339)$$

or, for very small step sizes,

$$\tau_n \simeq \frac{1}{\alpha\lambda_n}, \quad \alpha \ll 1. \quad (7.340)$$

The transient behavior of the weight vector follows easily. Pre-multiplying (7.331) by  $\mathbf{U}$ , using the unitary nature of  $\mathbf{U}$ , and adding  $\mathbf{w}(0)$ , we obtain

$$\mathbf{w}(K) = \mathbf{w}_o + \mathbf{U}\mathbf{v}(K). \quad (7.341)$$

The beam pattern at iteration  $K$  can be written as

$$\begin{aligned} B_K(\psi) &\stackrel{\Delta}{=} \mathbf{w}^H(K)\mathbf{v}_a(\psi) \\ &= \mathbf{w}_o^H\mathbf{v}_a(\psi) + \mathbf{v}^H(K)\mathbf{U}^H\mathbf{v}_a(\psi), \end{aligned} \quad (7.342)$$

where  $\mathbf{v}_a(\psi)$  is the array manifold vector. Using (7.335), we can write (7.342) as,

$$B_K(\psi) = B_o(\psi) + \sum_{n=1}^N v_n(0) (1 - \alpha\lambda_n)^K B_{eig,n}(\psi), \quad (7.343)$$

where

$$B_{eig,n}(\psi) = \Phi_n^H \mathbf{v}_a(\psi), \quad n = 1, 2, \dots, N, \quad (7.344)$$

is the  $n$ th eigenbeam. The steady state solution can also be written as a sum of eigenbeams. Replacing  $\mathbf{S}_x$  with its eigenvector expansion in (7.329) gives

$$\mathbf{w}_o = \sum_{n=1}^N \frac{1}{\lambda_n} \Phi_n \Phi_n^H \mathbf{p}. \quad (7.345)$$

Thus,

$$B_o(\psi) = \sum_{n=1}^N \left( \frac{\mathbf{p}^H \Phi_n}{\lambda_n} \right) B_{eig,n}(\psi). \quad (7.346)$$

The size of  $\alpha$  will be limited by the largest eigenvalue. The corresponding eigenbeam will converge with a time constant  $\alpha\lambda_{max}$ . However the eigenbeam corresponding to the smallest eigenvalue converges with a time constant of  $\alpha\lambda_{min}$ . If  $\lambda_{max}/\lambda_{min}$  is large, then the convergence time to the optimum pattern may be unacceptably long.

All of the discussion up to (7.340) is applicable to an arbitrary MMSE filtering problem. Before proceeding, it is important to show how we use steepest descent for beamforming. We first consider a representative example and then generalize the results.

#### Example 7.6.1

Consider an  $N$ -element array. The desired signal has power  $\sigma_s^2$  and arrives along a plane wave from  $\mathbf{v}_s$ . There is a single plane-wave interferer arriving along  $\mathbf{v}_I$  with power  $\sigma_I^2$ . There is additive white noise with variance  $\sigma_w^2$ .

Therefore, the input correlation matrix is

$$\mathbf{S}_x = \sigma_s^2 \mathbf{v}_s \mathbf{v}_s^H + \sigma_I^2 \mathbf{v}_I \mathbf{v}_I^H + \sigma_w^2 \mathbf{I}. \quad (7.347)$$

The input is,

$$\mathbf{X}(k) = \mathbf{v}_s F(k) + \mathbf{v}_I S_I(k) + \mathbf{N}(k), \quad (7.348)$$

so

$$\mathbf{p} \stackrel{\Delta}{=} \mathbf{S}_{xd} = E[\mathbf{X}(k) F^*(k)] = \mathbf{v}_s \sigma_s^2, \quad (7.349)$$

because the desired signal is uncorrelated with the interfering signal and the additive noise.

We first expand  $\mathbf{S}_x$  in an eigenvector expansion.

$$\mathbf{S}_x = \lambda_1 \Phi_1 \Phi_1^H + \lambda_2 \Phi_2 \Phi_2^H + \sum_{n=3}^N \lambda_n \Phi_n \Phi_n^H. \quad (7.350)$$

The first two eigenvectors are linear combinations of  $\mathbf{v}_s$  and  $\mathbf{v}_I$ . The remaining eigenvectors are the noise eigenvectors and are orthogonal to  $\Phi_1$  and  $\Phi_2$  and, therefore,  $\mathbf{v}_s$  and  $\mathbf{v}_I$ . We can rewrite (7.345) as

$$\mathbf{w}_o = \mathbf{S}_x^{-1} \mathbf{p}. \quad (7.351)$$

Substituting (7.349) and the inverse of (7.350) into (7.351) gives

$$\mathbf{w}_o = \frac{\sigma_s^2}{\lambda_1} \Phi_1 (\Phi_1^H \mathbf{v}_s) + \frac{\sigma_s^2}{\lambda_2} \Phi_2 (\Phi_2^H \mathbf{v}_s), \quad (7.352)$$

where we have used the orthogonality of  $\Phi_n, n = 3, \dots, N$  and  $\mathbf{v}_s$ . Assuming  $\mathbf{w}(0) = 0$ , then, using (7.352) in (7.333) gives,

$$\mathbf{v}(0) = -\mathbf{U}^H \left[ \frac{\sigma_s^2}{\lambda_1} \Phi_1 (\Phi_1^H \mathbf{v}_s) + \frac{\sigma_s^2}{\lambda_2} \Phi_2 (\Phi_2^H \mathbf{v}_s) \right], \quad (7.353)$$

so

$$v_1(0) = -\frac{\sigma_s^2}{\lambda_1} (\Phi_1^H \mathbf{v}_s), \quad (7.354)$$

$$v_2(0) = -\frac{\sigma_s^2}{\lambda_2} (\Phi_2^H \mathbf{v}_s), \quad (7.355)$$

and

$$v_n(0) = 0, \quad n = 3, \dots, N. \quad (7.356)$$

Thus, there are two natural modes in the steepest descent algorithm, and the algorithm behavior can be studied in a 2-D space (the signal-plus-interference subspace).

Note that the noise power enters into the beamformer through  $\lambda_1$  and  $\lambda_2$ ,

$$\lambda_1 = \lambda'_1 + \sigma_w^2, \quad (7.357)$$

$$\lambda_2 = \lambda'_2 + \sigma_w^2, \quad (7.358)$$

where  $\lambda'_1$  and  $\lambda'_2$  are the eigenvalues due to the signal and interference. Therefore the eigenvalue behavior will be a function of both the signal and interference characteristics (geometry and power levels) and the white noise level.

A more realistic initial condition is to assume

$$\mathbf{w}(0) = \frac{\mathbf{v}_s}{N}. \quad (7.359)$$

This weighting corresponds to uniform quiescent weighting. We consider this case in Example 7.6.2

#### Example 7.6.2 (continuation)

We consider the same model as in Example 7.6.1 with the initial weight-error vector given by (7.359). The optimum weight vector  $\mathbf{w}_o$  is still given by (7.352). The initial weight-error vector is

$$\mathbf{w}_e(0) = \frac{\mathbf{v}_s}{N} - \frac{\sigma_s^2}{\lambda_1} \Phi_1 (\Phi_1^H \mathbf{v}_s) - \frac{\sigma_s^2}{\lambda_2} \Phi_2 (\Phi_2^H \mathbf{v}_s), \quad (7.360)$$

and

$$\mathbf{v}(0) = \mathbf{U}^H \frac{\mathbf{v}_s}{N} - \mathbf{U}^H \left\{ \frac{\sigma_s^2}{\lambda_1} \Phi_1 (\Phi_1^H \mathbf{v}_s) + \frac{\sigma_s^2}{\lambda_2} \Phi_2 (\Phi_2^H \mathbf{v}_s) \right\}. \quad (7.361)$$

Then,

$$v_n(0) = \Phi_n^H \frac{\mathbf{v}_s}{N} - \Phi_n^H \left\{ \frac{\sigma_s^2}{\lambda_1} \Phi_1 (\Phi_1^H \mathbf{v}_s) + \frac{\sigma_s^2}{\lambda_2} \Phi_2 (\Phi_2^H \mathbf{v}_s) \right\}. \quad (7.362)$$

Using the orthogonality of the eigenvectors,

$$v_1(0) = \left( \frac{1}{N} - \frac{\sigma_s^2}{\lambda_1} \right) \Phi_1^H \mathbf{v}_s, \quad (7.363)$$

$$v_2(0) = \left( \frac{1}{N} - \frac{\sigma_s^2}{\lambda_2} \right) \Phi_2^H \mathbf{v}_s, \quad (7.364)$$

and

$$v_n(0) = 0, \quad n = 3, \dots, N. \quad (7.365)$$

Thus, once again there are two natural modes in the steepest descent algorithm. We only get a non-zero  $v_n(0)$  for  $n \geq 3$  when  $\mathbf{w}(0)$  has components outside of the  $\Phi_1^H, \Phi_2^H$  subspace. This behavior occurs if we use a quiescent weighting such as Dolph-Chebychev to obtain lower sidelobes in the quiescent beam pattern.

The results in Examples 7.6.1 and 7.6.2 generalize directly to the general case of a plane-wave desired signal and  $D$  interfering plane-wave signals. In this case, the steepest descent algorithm is operating in a  $D + 1$  subspace.

We next investigate the transient behavior of the MSE. We then return to Examples 7.6.1 and 7.6.2 and consider its behavior for various signal-and-interference environments.

The performance is determined by the transient behavior of the MSE. Using (7.320), (7.324), and (7.331), we have

$$\xi(K) = \xi_0 + \sum_{n=1}^N \lambda_n |v_n(K)|^2, \quad (7.366)$$

where  $\xi(K)$  is defined to be the MSE at sample  $K$ .<sup>17</sup> Substituting (7.335) into (7.366), we obtain

$$\xi(K) = \xi_0 + \sum_{n=1}^N \lambda_n (1 - \alpha \lambda_n)^{2K} |v_n(0)|^2, \quad (7.367)$$

and the transient MSE is

$$\begin{aligned} \xi_{sd}(K) &= \xi(K) - \xi_0 \\ &= \sum_{n=1}^N \lambda_n (1 - \alpha \lambda_n)^{2K} |v_n(0)|^2, \end{aligned} \quad (7.368)$$

where the subscript “*sd*” denotes steepest descent. The transient behavior of the error is a sum of geometric series, each one corresponding to a mode

---

<sup>17</sup>Note that  $\xi(K)$  is an ensemble average at iteration  $K$  and is not the same as the weighted residual error  $\xi(K)$  in (7.275).

of the algorithm. We illustrate the behavior with the same model as in Example 7.6.1.

**Example 7.6.3** (continuation)

We consider the same model as in Example 7.6.1. One can show that

$$\frac{\xi_o}{\sigma_s^2} = \frac{1 + N(\text{INR})}{1 + N(\text{SNR} + \text{INR}) + N^2(\text{SNR})(\text{INR})(1 - |\rho_{SI}|^2)} \quad (7.369)$$

where

$$\text{INR} = \frac{\sigma_I^2}{\sigma_w^2}, \quad (7.370)$$

and

$$\text{SNR} = \frac{\sigma_s^2}{\sigma_w^2}. \quad (7.371)$$

Using (7.354) and (7.355) in (7.368), we see that the normalized transient error is

$$\frac{\xi_{sd}(K)}{\sigma_s^2} = \frac{\sigma_s^2}{\lambda_1} (1 - \alpha\lambda_1)^{2K} |\Phi_1^H \mathbf{v}_s|^2 + \frac{\sigma_s^2}{\lambda_2} (1 - \alpha\lambda_2)^{2K} |\Phi_2^H \mathbf{v}_s|^2. \quad (7.372)$$

From the above discussion we see that the behavior of the algorithm will be determined by

- (i) The step size  $\alpha$ ;
- (ii) The eigenvalue spread  $\lambda_{max}/\lambda_{min}$  in  $\mathbf{S}_x$ .

We now consider two simple examples to illustrate the behavior. From Chapter 5, we know how the plane-wave spatial distribution will affect the eigenvalues. When looking at the results, it is important to remember that we are using ensemble averages. By contrast, in Sections 7.3 to 7.5, the recursions used time averages. In Section 7.7, we will use time averages and obtain a more valid comparison.

In the next two examples, we consider a standard 10-element linear array. We assume that there is a 10 dB SNR plane-wave signal arriving from broadside and interfering plane-wave signals arriving from various  $u_I$ . By varying the power of the interfering signal and its location we can see how the eigenvalue spread affects the algorithm performance. We use  $\mathbf{w}(0) = \mathbf{v}_s/N$  as the initial condition.

In each example we show two figures. The first is a representative beam pattern for various  $K$ , and the second is a plot of the transient MSE versus  $K$ .

We have chosen the parameters to show the effect of different eigenvalue spreads. We have used a scenario with a single signal to keep the plots reasonably simple.

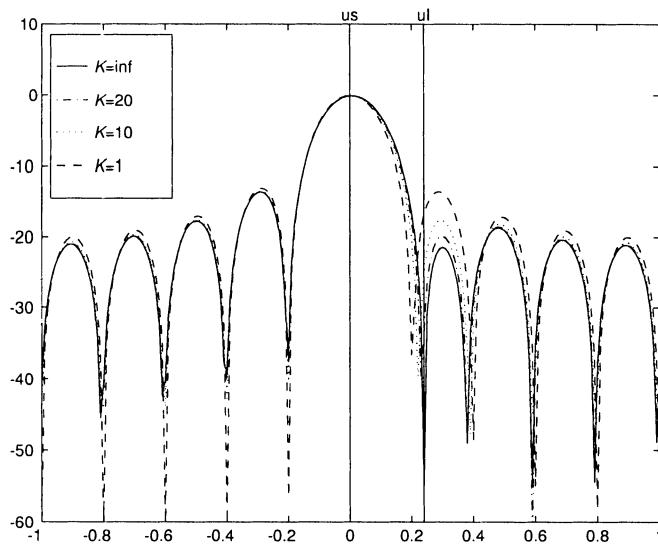


Figure 7.33 Steepest descent beamformer; beam patterns for various  $K$ ;  $u_s = 0$ ,  $SNR = 10$  dB,  $u_I = 0.24$ ,  $INR = 10$  dB,  $\mathbf{w}(0) = \mathbf{v}_s/N$ ,  $\alpha = 0.001$ .

#### Example 7.6.4

In this example, we use the following parameters:

$$\begin{aligned} SNR &= 10 \text{ dB}, & u_s &= 0, \\ INR &= 10 \text{ dB}, & u_I &= 0.24. \end{aligned}$$

The resulting eigenvalues are

$$\begin{aligned} \lambda_1 &= 117 \\ \lambda_2 &= 85. \end{aligned}$$

We use  $\alpha = 0.001$ .

In Figure 8.2, we show the beam patterns for various  $K$  for  $\mathbf{w}(0) = \mathbf{v}_s/N$ . In Figure 8.3, we show the transient mean square error  $\xi_T$ .

In this case the eigenvalue ratio is 1.376 and the two orthogonal modes behave in a similar manner.

The steepest descent beamformer has reached steady state by  $K = 30$ . The small eigenvalue spread is a favorable scenario for the SD algorithm.

#### Example 7.6.5

In this example, we use the following parameters:

$$\begin{aligned} u_s &= 0 & SNR &= 10 \text{ dB}, \\ u_{I1} &= 0.29 & INR_1 &= 20 \text{ dB}, \\ u_{I2} &= -0.45 & INR_2 &= 40 \text{ dB}. \end{aligned}$$

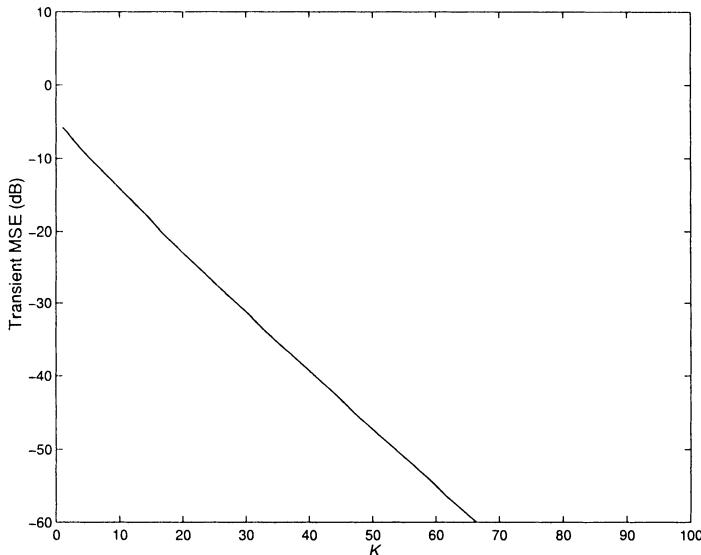


Figure 7.34 Steepest descent beamformer;  $\xi_{sd}(K)$  versus  $K$ .

The resulting eigenvalues are

$$\begin{aligned}\lambda_1 &= 1.0001 \times 10^5, \\ \lambda_2 &= 0.01 \times 10^5, \quad \lambda_3 = 0.0009 \times 10^5.\end{aligned}$$

We use  $\alpha = 4.9995 \times 10^{-6}$ .

In Figure 7.35, we show the beam patterns for various  $K$  for  $\mathbf{w}(0) = \mathbf{v}_s/N$ . In Figure 7.36, we show the transient mean square error  $\xi_T$ .

In this case, the dominant eigenvector is highly correlated with the array manifold vector for the 40 dB interferer at  $u_{I2} = -0.45$ , so it is nulled quickly. The convergence of the beam pattern at  $u_{I1} = 0.29$  is much slower and, by  $K = 100$ , the beam pattern has not started to form a null.

This completes our initial discussion of the steepest descent algorithm for solving the MMSE problem. We return to it in the next section in the context of generalized sidelobe canceller.

### 7.6.3 Steepest Decent: LCMP Beamformer

In this section, we derive a steepest descent algorithm to find the LCMP beamformer that we originally studied in Section 6.7. The result is due to Frost<sup>18</sup> [Fro72], and our derivation follows that reference.

---

<sup>18</sup>[Fro72] actually considers a gain-only constraint, but points out that the extension to multiple constraints is straightforward. He also considers the broadband model using a

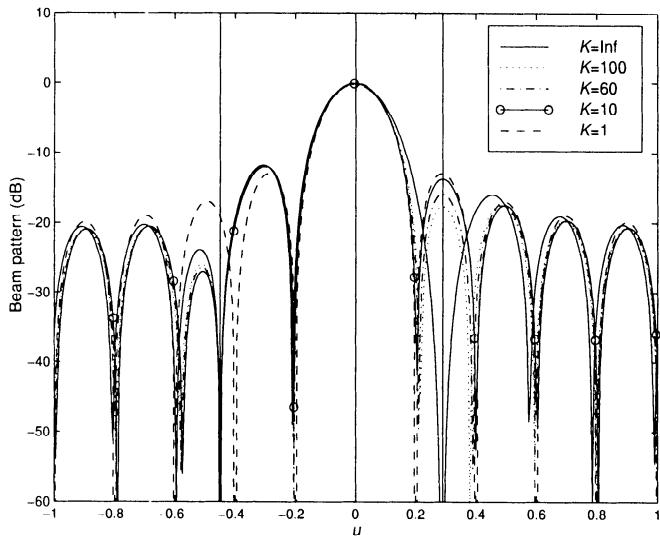


Figure 7.35 Steepest descent beamformer:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 20$  dB,  $u_{I2} = -0.45$ ,  $INR_2 = 40$  dB,  $\mathbf{w}(0) = \mathbf{v}_s/N$ ,  $\alpha = 4.9995 \times 10^{-6}$ ; beam patterns for various  $K$ .

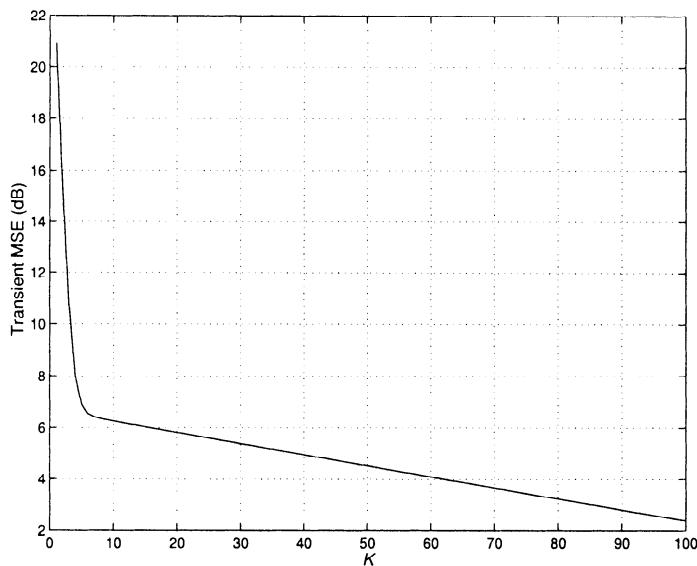


Figure 7.36 Steepest descent beamformer:  $\xi_{sd}(K)$  versus  $K$ .

From (6.353), we want to minimize

$$J \triangleq \mathbf{w}^H \mathbf{S}_x \mathbf{w} + [\mathbf{w}^H \mathbf{C} - \mathbf{g}^H] \boldsymbol{\lambda} + \boldsymbol{\lambda}^H [\mathbf{C}^H \mathbf{w} - \mathbf{g}]. \quad (7.373)$$

The gradient with respect to  $\mathbf{w}^H$  is

$$\nabla_{\mathbf{w}^H} = \mathbf{S}_x \mathbf{w} + \mathbf{C} \boldsymbol{\lambda}. \quad (7.374)$$

In Section 7.6.2, we set the gradient equal to zero and found that

$$\mathbf{w}_o = \mathbf{S}_x^{-1} \mathbf{C} [\mathbf{C}^H \mathbf{S}_x \mathbf{C}]^{-1} \mathbf{g} \quad (7.375)$$

(see (6.358)).

To find  $\mathbf{w}$  using a steepest descent algorithm, we follow the same procedure as in Section 7.6.2.

$$\begin{aligned} \mathbf{w}(K) &= \mathbf{w}(K-1) - \alpha \nabla_{\mathbf{w}^H} (J(K-1)) \\ &= \mathbf{w}(K-1) - \alpha [\mathbf{S}_x \mathbf{w}(K-1) + \mathbf{C} \boldsymbol{\lambda}(K-1)]. \end{aligned} \quad (7.376)$$

We require  $\mathbf{w}(K)$  to satisfy the constraint.

$$\mathbf{g} = \mathbf{C}^H \mathbf{w}(K) = \mathbf{C}^H \mathbf{w}(K-1) - \alpha \mathbf{C}^H \mathbf{S}_x \mathbf{w}(K-1) - \alpha \mathbf{C}^H \mathbf{C} \boldsymbol{\lambda}(K-1). \quad (7.377)$$

Solving for  $\boldsymbol{\lambda}(K-1)$  and substituting into (7.376) gives

$$\begin{aligned} \mathbf{w}(K) &= \mathbf{w}(K-1) - \alpha \left( \mathbf{I} - \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{C}^H \right) \mathbf{S}_x \mathbf{w}(K-1) \\ &\quad + \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} [\mathbf{g} - \mathbf{C}^H \mathbf{w}(K-1)]. \end{aligned} \quad (7.378)$$

Frost [Fro72] points out that one should not assume that the term in the last brackets is zero. By retaining it we improve the numerical stability compared to previous algorithms ([Ros60], [BOBH69]).

From (6.367),

$$\mathbf{w}_q = \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{g}, \quad (7.379)$$

where

$$\mathbf{P}_c^\perp = \left[ \mathbf{I} - \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{C}^H \right]. \quad (7.380)$$

---

tapped-delay line. We study that case in Section 7.11.

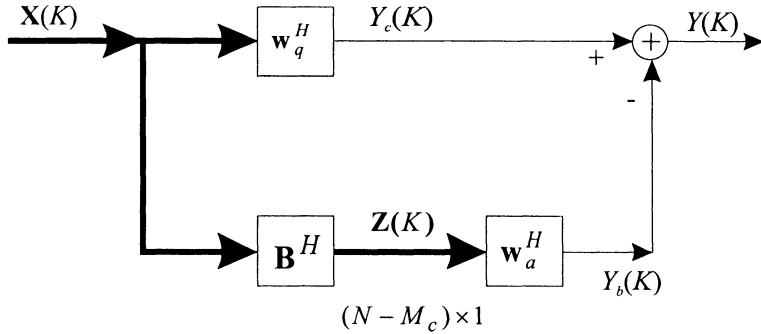


Figure 7.37 Generalized sidelobe canceller.

We can write (7.378) as

$$\mathbf{w}(K) = \mathbf{P}_c^\perp [\mathbf{w}(K-1) - \alpha \mathbf{S}_x \mathbf{w}(K-1)] + \mathbf{w}_q, \quad (7.381)$$

or

$$\boxed{\mathbf{w}(K) = \mathbf{P}_c^\perp [\mathbf{I} - \alpha \mathbf{S}_x] \mathbf{w}(K-1) + \mathbf{w}_q}, \quad (7.382)$$

with

$$\boxed{\mathbf{w}(0) = \mathbf{w}_q}. \quad (7.383)$$

Before we analyze the error behavior of the steepest descent LCMP algorithm we want to develop the generalized sidelobe canceller version of the algorithm.

The GSC implementation was developed in Section 7.4.4 and is shown in Figure 7.37.

Recall that

$$\mathbf{B}^H \mathbf{C} = \mathbf{0}. \quad (7.384)$$

In addition, we assume that the columns of  $\mathbf{B}$  are orthonormal,

$$\mathbf{B}^H \mathbf{B} = \mathbf{I}. \quad (7.385)$$

We denote the total weight vector in Figure 7.37 by  $\mathbf{w}_{gsc}$ ,

$$\mathbf{w}_{gsc} = \mathbf{w}_q - \mathbf{B} \mathbf{w}_a. \quad (7.386)$$

The steepest descent algorithm for  $\mathbf{w}_a$  is just the MMSE-SD algorithm in (7.326) with the desired signal equal to  $Y_c(k)$ . The result is

$$\boxed{\mathbf{w}_a(K) = \mathbf{w}_a(K-1) + \alpha [\mathbf{p}_B - \mathbf{S}_z \mathbf{w}_a(K-1)]}, \quad (7.387)$$

where

$$\mathbf{S}_z \triangleq \mathbf{B}^H \mathbf{S}_x \mathbf{B}, \quad (7.388)$$

and

$$\mathbf{p}_B \triangleq E[\mathbf{Z}D^*] = \mathbf{B}^H \mathbf{S}_x \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{g}. \quad (7.389)$$

One can show that if we use identical initial conditions in the direct form LCMP-SD beamformer and the GSC version of the LCMP-SD beamformer,

$$\mathbf{w}_{gsc}(0) = \mathbf{w}_{lcmpl}(0) = \mathbf{w}_q, \quad (7.390)$$

which implies

$$\mathbf{w}_a(0) = \mathbf{0}, \quad (7.391)$$

then the behavior of the two SD algorithms will be identical ((7.385) must be satisfied).

We can analyze the weight-error vector of the LCMP-SD beamformers in exactly the same manner as Section 7.6.2. Define

$$\mathbf{w}_{ae}(K) = \mathbf{w}_a(K) - \mathbf{w}_{ao}, \quad (7.392)$$

where  $\mathbf{w}_{ao}$  is the optimum weight vector given by

$$\mathbf{p}_B = \mathbf{S}_z \mathbf{w}_{ao}. \quad (7.393)$$

Substituting (7.392) into (7.387) gives

$$\mathbf{w}_{ae}(K) = [\mathbf{I} - \alpha \mathbf{S}_z] \mathbf{w}_{ae}(K-1), \quad (7.394)$$

which has the same form as (7.328). To analyze the behavior, rotate into an orthogonal coordinate system,

$$\mathbf{S}_z = \mathbf{B}^H \mathbf{S}_x \mathbf{B} = \mathbf{U}_c \boldsymbol{\lambda}_c \mathbf{U}_c^H. \quad (7.395)$$

Then

$$\mathbf{v}_c(K) = \mathbf{U}_c^H \mathbf{w}_e(K) = [\mathbf{I} - \alpha \boldsymbol{\Lambda}_c] \mathbf{v}_c(K-1). \quad (7.396)$$

If there are  $M$  constraints, then  $\mathbf{S}_z$  will have  $N-M$  eigenvalues and eigenvectors. One can show that

$$\lambda_{min} \leq \lambda_{B,min} \leq \lambda_{B,max} \leq \lambda_{max} \quad (7.397)$$

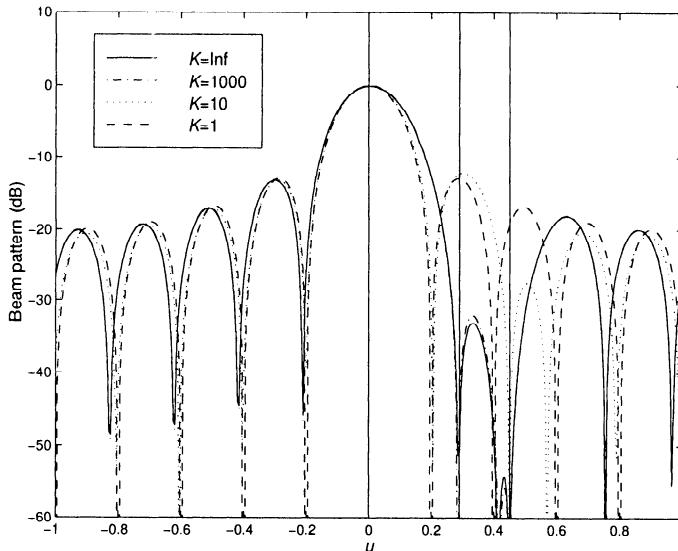


Figure 7.38 MPDR-GSC-SD beamformer, beam patterns for various  $K$ ;  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 10$  dB,  $u_{I2} = 0.45$ ,  $INR_2 = 30$  dB,  $\alpha = 0.5/\lambda_{max}$ .

(see [Fro72]). Therefore the convergence depends on fewer modes, and the rates are between the slowest and fastest MMSE modes.

We consider a simple example to illustrate the behavior. The eigenvalues of  $\mathbf{S}_z$  determine the convergence rate, so we need a scenario with multiple interferers in order to have an interesting example.

#### Example 7.6.6

Consider a SLA10 array and the same signal-and-interference model as in Example 7.4.1. The desired signal arrives from  $u_s = 0$  with an  $SNR = 10$  dB. Two uncorrelated interferers arrive from  $u_I = 0.29$  and  $u_I = 0.45$ . The  $INR$  for the interferer at 0.29 is 10 dB. The  $INR$  for the interferer at 0.45 is varied from 10 dB to 60 dB in 10 dB steps. We implement an MPDR beamformer with a single distortionless constraint at  $u_s = 0$ . The initial condition is  $\mathbf{w}_a = 0$  and  $\mathbf{w}_q = \mathbf{v}_s/N$ . We use  $\alpha = 0.5/\lambda_{max}$ .

In Figure 7.38, we show representative beam patterns for various  $K$  with  $INR_2 = 30$  dB. We see interferer 2 is nulled by  $K = 10$ , but interferer 1 is not nulled until  $K = 1000$ . In Figure 7.39, we plot the  $SINR_o$  versus  $K$ . As the eigenvalue spread increases, the convergence to steady state slows dramatically.

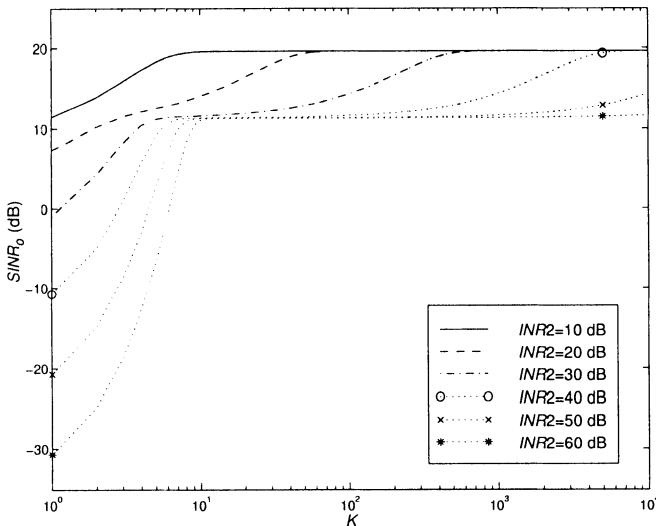


Figure 7.39 MPDR-GSC-SD beamformer,  $SINR_o$  versus  $K$ ;  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 10$  dB,  $u_{I2} = 0.45$ ,  $INR_2 = 10$  dB,  $\dots$ ,  $60$  dB,  $\alpha = 0.5/\lambda_{max}$ .

#### 7.6.4 Summary

In this section, we have discussed the steepest descent algorithm for the cases of most interest in adaptive beamforming, the MMSE beamformer and the LCMP beamformer. We have demonstrated that they converge to the optimum beamformer if the step size  $\alpha$  is chosen appropriately.

As we pointed out in the introduction, there are other deterministic gradient algorithms. Various Newton-type algorithms are most widely used. A discussion of them is available in Chapter 4 of [WS85] in the context of adaptive filtering.

In the next section we develop the stochastic gradient version of these algorithms and analyze their performance.

## 7.7 LMS Algorithms

In this section we develop the least mean-square (LMS) algorithm. It was originated by Widrow and his colleagues at Stanford University (e.g., [WS85], [WMGG67]). We develop two versions of unconstrained MMSE-LMS algo-

rithms and two versions of linearly constrained LMS algorithms. The first unconstrained algorithm is Widrow's original algorithm. This algorithm has been the foundation for a significant amount of adaptive array research and implementation in the last three decades. The second unconstrained algorithm in the LMS category is due to Griffiths [Gri69], and is referred to as the modified LMS algorithm or the Griffiths algorithm. Both algorithms are based on minimizing the MSE using gradient techniques. The two algorithms differ in the *a priori* knowledge they assume about the signal environment.

Both algorithms are stochastic gradient algorithms, as contrasted to the deterministic steepest descent algorithm in Section 7.6. Their primary advantage is computational simplicity. The computations behave as  $O(N)$ , as contrasted to  $O(N^2)$  of the QRD-RLS algorithms. Their primary disadvantage is a slower convergence in a complicated signal and interference environment.

The LMS algorithm is discussed in a number of articles and textbooks. Early articles include Widrow et al. [WMGG67], Griffiths [Gri69], and Frost [Fro72]. A recent tutorial article is Glentis et al. [GBT99], which has an extensive reference list. Book references include Compton [Com88], Monzingo and Miller [MM80], Widrow and Stearns [WS85], Sibul [Sib87], and Haykin [Hay96].

In Section 7.7.1, we derive the LMS algorithms. In Section 7.7.2, we study the statistical properties of the LMS algorithms. In Section 7.7.3, we demonstrate the algorithm behavior for several interesting cases. In Section 7.7.4, we consider LMS algorithms with quadratic constraints. In Section 7.7.5, we summarize our results.

### 7.7.1 Derivation of the LMS Algorithms

The LMS algorithm is the stochastic version of the steepest descent algorithm. In this section, we derive several versions of LMS algorithms.

#### 7.7.1.1 Widrow LMS algorithm

The first LMS algorithm of interest is due to Widrow et al. [WH60], [WMGG67], and is the stochastic version of the steepest descent algorithm in (7.322).

The gradient in the SD algorithm was given by (7.322), which we repeat

$$\nabla \xi_{\mathbf{w}^H} = -\mathbf{p} + \mathbf{S}_x \mathbf{w}. \quad (7.398)$$

In the LMS algorithm, we must estimate  $\mathbf{p}$  and  $\mathbf{S}_x$ .

A simple choice for estimates is

$$\hat{\mathbf{p}}(K) = \mathbf{X}(K)D^*(K), \quad (7.399)$$

and

$$\hat{\mathbf{S}}_{\mathbf{x}} = \mathbf{X}(K)\mathbf{X}^H(K). \quad (7.400)$$

In other words, the instantaneous values are used as estimates. Better estimates could be obtained by time averaging,

$$\hat{\mathbf{p}}(K) = \frac{1}{N_I} \sum_{k=K-N_I}^K \mathbf{X}(k)D^*(k), \quad (7.401)$$

and

$$\hat{\mathbf{S}}_{\mathbf{x}} = \frac{1}{N_I} \sum_{k=K-N_I}^K \mathbf{X}(k)\mathbf{X}^H(k). \quad (7.402)$$

Use of (7.401) and (7.402) leads to an LMS algorithm that is referred to as a sliding window LMS (SW-LMS) algorithm in the literature.

We could also use exponential weighting in the averages, as in Sections 7.3 and 7.4.

We use (7.399) and (7.400) in our present discussion. We find that the LMS algorithm provides an averaging effect. Then, the estimate of the gradient is

$$\hat{\nabla}\xi(K) = -\mathbf{X}(K)D^*(K) + \mathbf{X}(K)\mathbf{X}^H(K)\hat{\mathbf{w}}(K), \quad (7.403)$$

and the LMS algorithm is

$$\hat{\mathbf{w}}(K) = \hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K) \left[ D^*(K) - \mathbf{X}^H(K)\hat{\mathbf{w}}(K-1) \right]. \quad (7.404)$$

We use the notation  $\hat{\mathbf{w}}(K)$  because the algorithm is using an estimate of the gradient instead of the actual gradient. When we write the algorithm as in (7.404), it is an open-loop algorithm and its structure is similar to the RLS algorithm in Section 7.4. We will discuss the similarities in more detail in Section 7.7.1.5.

We can also write (7.404) as

$$\begin{aligned} \hat{\mathbf{w}}(K) &= \hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K) \left[ D^*(K) - \tilde{Y}_p^*(K) \right] \\ &= \hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K)e_p^*(K), \end{aligned} \quad (7.405)$$

where

$$\tilde{Y}_p^*(K) \stackrel{\triangle}{=} \mathbf{X}^H(K)\hat{\mathbf{w}}(K-1). \quad (7.406)$$

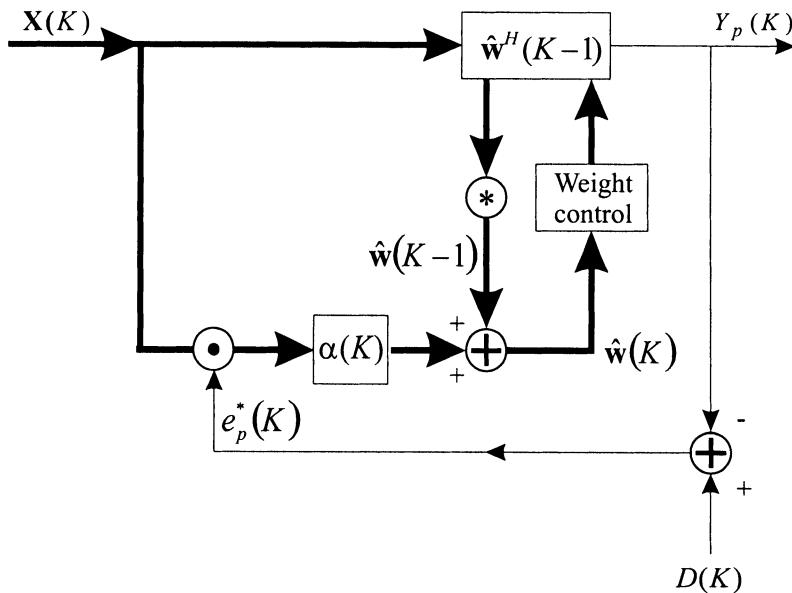


Figure 7.40 Block diagram of LMS algorithm.

A block diagram is shown in Figure 7.40.

If we were dealing with infinite precision arithmetic and perfect components, then (7.404) and (7.405) are identical. However, in an actual system there will be errors, so the LMS algorithm uses the actual output  $\hat{Y}_p^*(K)$  shown in Figure 7.40 and operates as a closed-loop system. The closed-loop operation tends to compensate for errors.

The initial condition,  $\hat{\mathbf{w}}(0)$ , will depend on the model. Common choices are  $\hat{\mathbf{w}}(0) = \mathbf{0}$ ,  $\hat{\mathbf{w}}(0) = \mathbf{v}_s/N$ , or  $\hat{\mathbf{w}}(0) = \mathbf{w}_q$ .

At each iteration,  $\mathbf{X}(K)$ ,  $D(K)$ , and  $\hat{\mathbf{w}}(K)$  are required. The estimates at a particular time may have large variances, and they introduce what is referred to as gradient noise. The result in (7.404) is the complex narrowband version of Widrow's original LMS algorithm (see [WMB75]).

Just as in the RLS case, introducing diagonal loading can provide improved performance in many cases. To include diagonal loading, modify (7.404) to

$$\begin{aligned}\hat{\mathbf{w}}(K) &= \hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K)D^*(K) \\ &\quad - \alpha(K) \left\{ \left[ \sigma_L^2 \mathbf{I} + \mathbf{X}(K)\mathbf{X}^H(K) \right] \hat{\mathbf{w}}(K-1) \right\},\end{aligned}\quad (7.407)$$

which can be rewritten as,

$$\hat{\mathbf{w}}(K) = \left(1 - \alpha(K)\sigma_L^2\right) \hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K) \left[D^*(K) - \tilde{Y}_p^*(K)\right], \quad (7.408)$$

or

$$\boxed{\hat{\mathbf{w}}(K) = \beta_L(K)\hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K) \left[D^*(K) - \tilde{Y}_p^*(K)\right]}, \quad (7.409)$$

where

$$\beta_L(K) = 1 - \alpha(K)\sigma_L^2. \quad (7.410)$$

We can also write (7.409) as

$$\hat{\mathbf{w}}(K) = \beta_L(K)\hat{\mathbf{w}}(K-1) + \alpha(K)\mathbf{X}(K)e_p^*(K). \quad (7.411)$$

We can also let the diagonal loading vary as a function of  $K$  by replacing  $\sigma_L^2$  with  $\sigma_L^2(K)$ .

Due to its simplicity, the LMS algorithm can also be implemented using analog techniques. Analog implementations are discussed in [Com88].

### 7.7.1.2 Griffiths LMS algorithm

A second version of the LMS algorithm is due to Griffiths ([Gri68], [Gri69]), and is referred to as the modified LMS algorithm, the Griffiths algorithm, or the steered direction algorithm. It assumes that

$$\mathbf{S}_{xd^*} = E[\mathbf{X}(k)D^*(k)], \quad (7.412)$$

is known. In the narrowband case,

$$\mathbf{X}(k) = \mathbf{v}_s F(k) + \mathbf{N}(k), \quad (7.413)$$

and

$$D^*(k) = F^*(k), \quad (7.414)$$

so

$$\mathbf{S}_{xd^*} = \sigma_s^2 \mathbf{v}_s. \quad (7.415)$$

Thus, the Griffiths algorithm assumes the signal direction of arrival and the signal power are known. However, the desired signal  $D(k)$  is not required.

The modified LMS algorithm is

$$\boxed{\hat{\mathbf{w}}(K) = \hat{\mathbf{w}}(K-1) + \alpha(K) \left[ \sigma_s^2 \mathbf{v}_s - \mathbf{X}(K) \tilde{Y}_p^*(K) \right].} \quad (7.416)$$

The result in (7.416) is due to Griffiths [Gri69]. It is important to note that, although  $\mathbf{v}_s$  appears in (7.416), it is not imposing a hard constraint.

If we utilize diagonal loading, (7.416) becomes

$$\hat{\mathbf{w}}(K) = \beta_L(K)\hat{\mathbf{w}}(K-1) + \alpha(K) \left[ \sigma_s^2 \mathbf{v}_s - \mathbf{X}(K) \tilde{Y}_p^*(K) \right]. \quad (7.417)$$

### 7.7.1.3 Frost LMS algorithm

The third version of the LMS algorithm imposes a linear constraint. This algorithm is the stochastic gradient version of the steepest descent algorithm in Section 7.6.3. Using (7.400) in (7.382) gives

$$\hat{\mathbf{w}}(K) = \mathbf{P}_{\mathbf{c}}^\perp \left[ \hat{\mathbf{w}}(K-1) - \alpha(K) \mathbf{X}(K) \tilde{Y}_p^*(K) \right] + \mathbf{w}_q, \quad (7.418)$$

where

$$\tilde{Y}_p(K) = \hat{\mathbf{w}}^H(K-1) \mathbf{X}(K), \quad (7.419)$$

and

$$\hat{\mathbf{w}}(0) = \mathbf{w}_q. \quad (7.420)$$

For the simple case of the MPDR beamformer,

$$\mathbf{w}_q = \frac{1}{N} \mathbf{v}_s, \quad (7.421)$$

and

$$\mathbf{P}_{\mathbf{c}}^\perp = \mathbf{I} - \mathbf{v}_s \left( \mathbf{v}_s^H \mathbf{v}_s \right)^{-1} \mathbf{v}_s^H. \quad (7.422)$$

For the general LCMP case,

$$\mathbf{w}_q = \mathbf{C} \left( \mathbf{C}^H \mathbf{C} \right)^{-1} \mathbf{g}, \quad (7.423)$$

and

$$\mathbf{P}_{\mathbf{c}}^\perp = \mathbf{I} - \mathbf{C} \left( \mathbf{C}^H \mathbf{C} \right)^{-1} \mathbf{C}^H. \quad (7.424)$$

This is the narrowband complex version of the linearly constrained beamformer which was originally derived by Frost [Fro72]. Frost derived the real wideband version for a gain-only (i.e., distortionless) constraint, but indicated the extension to multiple constraints would be straightforward.

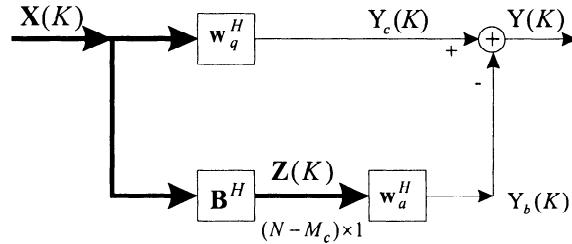


Figure 7.41 Generalized sidelobe canceller.

#### 7.7.1.4 Generalized sidelobe canceller LMS algorithm

The fourth version of the LMS algorithm is the GSC version, which is shown in Figure 7.41. Recall that  $Y_c(k)$  corresponds to  $D(k)$  in the MMSE algorithm. Adapting (7.405),

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \alpha(K) \mathbf{Z}(K) e_{pa}^*(K), \quad (7.425)$$

where

$$\begin{aligned} e_{pa}(K) &= Y_c(K) - \hat{\mathbf{w}}_a^H(K-1) \mathbf{Z}(K) \\ &= Y_c(K) - \tilde{Y}_{bp}(K). \end{aligned} \quad (7.426)$$

The total weight vector is

$$\hat{\mathbf{w}}(K) = \mathbf{w}_q - \mathbf{B} \hat{\mathbf{w}}_a(K), \quad (7.427)$$

and

$$Y(K) = \hat{\mathbf{w}}^H(K) \mathbf{X}(K). \quad (7.428)$$

The initial condition is

$$\hat{\mathbf{w}}_a(0) = \mathbf{0}. \quad (7.429)$$

The result in (7.425)–(7.429) is the narrowband complex version of the GSC beamformer derived by Griffiths and Jim [GJ82].

If we choose  $\mathbf{B}$  so that

$$\mathbf{B}^H \mathbf{B} = \mathbf{I}, \quad (7.430)$$

the adaptive performance of GSC implementation will be identical to the adaptive performance of direct form implementation. Diagonal loading can be included.

### 7.7.1.5 Comparison of RLS and LMS algorithms

It is useful to compare the LMS algorithm in (7.404) with the RLS algorithm of Section 7.4.<sup>19</sup>

From (7.189),

$$\hat{\mathbf{w}}_{rls}(K) = \hat{\mathbf{w}}_{rls}(K-1) + \Phi^{-1}(K)\mathbf{X}(K) \left[ D^*(K) - \mathbf{X}^H(K)\hat{\mathbf{w}}_{rls}(K-1) \right] \quad (7.431)$$

and, repeating (7.404)

$$\hat{\mathbf{w}}_{lms}(K) = \hat{\mathbf{w}}_{lms}(K-1) + \alpha(K)\mathbf{X}(K) \left[ D^*(K) - \mathbf{X}^H(K)\hat{\mathbf{w}}_{lms}(K-1) \right]. \quad (7.432)$$

We see that the difference between the two algorithms, as expressed by (7.431) and (7.432), is that the error term is multiplied by  $\Phi^{-1}(K)$  in the RLS algorithm and  $\alpha(K)\mathbf{I}$  in the LMS algorithm.

The multiplication by  $\Phi^{-1}(K)$  causes each mode to converge at the same rate in the RLS algorithm so that the eigenvalue spread does not limit the convergence rate. The disadvantage is that computing  $\Phi^{-1}(K)$  requires more computation. Recall that

$$\mathbf{g}(K) = \mathbf{P}(K)\mathbf{X}(K) = \Phi^{-1}(K)\mathbf{X}(K), \quad (7.433)$$

is the Kalman gain.

Our approach to deriving the LMS algorithm started with the deterministic steepest descent algorithm in Section 7.6. We replaced the deterministic quantities with estimated quantities to obtain the LMS algorithm. If we had started with a deterministic Newton algorithm and replaced the deterministic quantities with estimated quantities, we would have obtained the **LMS/Newton algorithm** or **LMSN algorithm**. This approach is discussed in Chapters 4 and 8 of [WS85]. Their discussion leads to an algorithm referred to as the **sequential regression algorithm (SER algorithm)** which they attribute to Graupe [Gra72], Ahmed et al. [ASHP77], Parikh and Ahmed [PA78], Ahmed et al. [AHUS79], and Lee [Lee66]. However, if the various parameters are chosen appropriately, the algorithm is identical to the RLS algorithm. The relation between the two algorithms is also discussed in Diniz et al. [DCA95].

In the next section, we derive the statistical properties of the LMS algorithms derived in Sections 7.7.1.1–7.7.1.4. Then, in Section 7.7.3, we simulate the algorithms for typical scenarios.

---

<sup>19</sup>The relationship between the RLS and LMS algorithms is discussed in a number of texts (e.g., Section 6.2.4 of [PRLN92] or Sections 13.6 and 13.7 of [Hay96]).

### 7.7.2 Performance of the LMS Algorithms

In order to understand the behavior of the LMS algorithms, we consider the following six questions:

1. Under what condition, and in what sense, does the weight vector converge to the optimum weight vector?
2. Assuming the weight vector converges, what is its transient behavior? How long does it take to converge?
3. As we are dealing with an estimate of the gradient, there may be some residual error in the weight vector. How large is this error and what does it depend on?
4. How complex is the computation required to implement the algorithm?
5. How is the desired signal derived?
6. How does the system behave for some typical examples?

We discuss the first five questions in this section and look at examples in Section 7.7.3. Most of these questions were answered in the original paper by Widrow and his colleagues [WMGG67]. Haykin [Hay96] has added several new arguments that we will utilize. There is a good tutorial discussion of the LMS algorithm and other algorithms in Glentis et al. [GBT99]. The article also contains extensive references.

In our discussion, we assume the input vectors  $\mathbf{X}(1), \mathbf{X}(2), \dots, \mathbf{X}(K)$  are statistically independent. We assume that  $\mathbf{X}(K)$  is statistically independent of all previous samples of the desired response  $D(1), \dots, D(K-1)$ . At time  $K$ , the desired response  $D(K)$  is dependent on  $\mathbf{X}(K)$ , but it is statistically independent of all previous samples of the desired response. The input  $\mathbf{X}(K)$  and the desired response  $D(K)$  are assumed to be jointly Gaussian for all  $K$ .

#### 7.7.2.1 Weight-error vector mean

We define the weight-error vector in the LMS algorithm as

$$\hat{\mathbf{w}}_e(K) = \hat{\mathbf{w}}(K) - \mathbf{w}_o. \quad (7.434)$$

Using (7.434) in (7.404) gives

$$\hat{\mathbf{w}}_e(K) = \hat{\mathbf{w}}_e(K-1) + \alpha(K)\mathbf{X}(K) \left[ D^*(K) - \mathbf{X}^H(K)(\hat{\mathbf{w}}_e(K-1) + \mathbf{w}_o) \right]. \quad (7.435)$$

Taking the expectation of both sides, using the independence assumption, and using (7.323), we obtain

$$E[\hat{\mathbf{w}}_e(K)] = [\mathbf{I} - \alpha(K)\mathbf{S}_x] E[\hat{\mathbf{w}}_e(K-1)]. \quad (7.436)$$

We see that (7.436) and (7.328) have identical form. Since  $E[\hat{\mathbf{w}}_e(0)] = \mathbf{w}_e(0)$ ,

$$E[\hat{\mathbf{w}}_e(K)] = \mathbf{w}_e(K), \quad (7.437)$$

so that the mean of the LMS weight-error vector is the weight-error vector in the steepest descent algorithm.

We also observe that we could obtain the same result for the Griffiths LMS algorithm (7.416).

### 7.7.2.2 Weight-error correlation matrix

The correlation matrix of  $\hat{\mathbf{w}}_e(K)$  is<sup>20</sup>

$$\mathbf{R}_e(K) = E[\hat{\mathbf{w}}_e(K)\hat{\mathbf{w}}_e^H(K)]. \quad (7.438)$$

To analyze (7.438), we rewrite (7.435) as

$$\begin{aligned} \hat{\mathbf{w}}_e(K) &= \hat{\mathbf{w}}_e(K-1) + \alpha(K)\mathbf{X}(K) [e_o^*(K) - \mathbf{X}^H(K)\hat{\mathbf{w}}_e(K-1)] \\ &= [\mathbf{I} - \alpha(K)\mathbf{X}(K)\mathbf{X}^H(K)]\hat{\mathbf{w}}_e(K-1) + \alpha(K)\mathbf{X}^H(K)e_o(K), \end{aligned} \quad (7.439)$$

where

$$e_o(K) = D(K) - \mathbf{w}_o^H\mathbf{X}(K), \quad (7.440)$$

is the estimation error using the MMSE weight vector. If  $\alpha(K)$  is small, then the coefficient of  $\hat{\mathbf{w}}_e(K-1)$  in (7.439) is close to  $\mathbf{I}$  and the behavior of  $\hat{\mathbf{w}}_e(K-1)$  may be approximated by the stochastic difference equation:<sup>21</sup>

$$\hat{\mathbf{w}}_e(K) = [\mathbf{I} - \alpha(K)\mathbf{S}_x]\hat{\mathbf{w}}_e(K-1) + \alpha(K)\mathbf{X}^H(K)e_o(K). \quad (7.441)$$

Substituting (7.441) into (7.438) gives

$$\begin{aligned} \mathbf{R}_e(K) &= [\mathbf{I} - \alpha(K)\mathbf{S}_x]\mathbf{R}_e(K-1)[\mathbf{I} - \alpha(K)\mathbf{S}_x] \\ &\quad + \alpha^2(K)E[e_o^*(K)\mathbf{X}(K)\mathbf{X}^H(K)e_o(K)]. \end{aligned} \quad (7.442)$$

---

<sup>20</sup>This discussion is similar to Chapter 9 of [Hay96].

<sup>21</sup>This approach is due to Kushner [Kus84] and is described on p. 396 of [Hay96].

Using the moment factorization of jointly Gaussian random variables and the independence assumption,

$$E \left[ e_o^*(K) \mathbf{X}(K) \mathbf{X}^H(K) e_o(K) \right] = \xi_o \mathbf{S}_x, \quad (7.443)$$

where

$$\xi_o = E \left[ |e_o(K)|^2 \right], \quad (7.444)$$

is the MSE using the optimum weight vector. Then, (7.442) can be written as

$$\mathbf{R}_e(K) = [\mathbf{I} - \alpha(K) \mathbf{S}_x] \mathbf{R}_e(K-1) [\mathbf{I} - \alpha(K) \mathbf{S}_x] + \alpha^2(K) \xi_0 \mathbf{S}_x. \quad (7.445)$$

We can use (7.445) to compute the excess MSE. The LMS error can be written as

$$e_{lms}(K) = D(K) - \hat{\mathbf{w}}^H(K) \mathbf{X}(K). \quad (7.446)$$

Using (7.434) and (7.440) in (7.446) gives

$$e_{lms}(K) = e_o(K) - \hat{\mathbf{w}}_e^H(K) \mathbf{X}(K). \quad (7.447)$$

The mean-square LMS error at iteration  $K$  is

$$\begin{aligned} \xi_{lms}(K) &= E \left[ |e_o(K) - \hat{\mathbf{w}}_e^H(K) \mathbf{X}(K)|^2 \right] \\ &= \xi_o + E \left[ \hat{\mathbf{w}}_e^H(K) \mathbf{X}(K) \mathbf{X}^H(K) \hat{\mathbf{w}}_e(K) \right] \\ &= \xi_o + \text{tr} \left\{ E \left[ \mathbf{X}(K) \mathbf{X}^H(K) \right] E \left[ \hat{\mathbf{w}}_e(K) \hat{\mathbf{w}}_e^H(K) \right] \right\} \\ &= \xi_o + \text{tr} \left\{ \mathbf{S}_x \mathbf{R}_e(K) \right\}. \end{aligned} \quad (7.448)$$

Now define an excess MSE,

$$\xi_{ex}(K) = \xi_{lms}(K) - \xi_o = \text{tr} \left\{ \mathbf{S}_x \mathbf{R}_e(K) \right\}. \quad (7.449)$$

The matrix inside the trace is positive definite so, at each value of  $K$ , the excess error is positive. The  $\mathbf{R}_e(K)$  term in (7.448) can be evaluated using (7.445). Using results by Macchi [Mac95], Haykin (pp. 399–402 of [Hay96]) analyzes the transient behavior. As  $K \rightarrow \infty$ , the excess MSE approaches a constant value if

$$0 < \alpha(K) < \frac{2}{\lambda_{max}}, \quad (7.450)$$

where  $\lambda_{max}$  is the largest eigenvalue of  $\mathbf{S}_x$ . If (7.450) is satisfied, the steady state excess error is

$$\xi_{ex}(\infty) = \xi_0 \sum_{i=1}^N \frac{\alpha(K)\lambda_i}{2 - \alpha(K)\lambda_i} \quad (7.451)$$

(e.g., pp. 397–400 of [Hay96]). The misadjustment is defined as the ratio of the excess error  $\xi_{ex}(\infty)$  to  $\xi_0$ :

$$\mathcal{M} = \frac{\xi_{ex}(\infty)}{\xi_0} = \sum_{i=1}^N \frac{\alpha(K)\lambda_i}{2 - \alpha(K)\lambda_i} \quad (7.452)$$

(e.g., p. 400 of [Hay96] or [WMGG67]).

In their original work, Widrow et al. [WMGG67] provide several useful approximations.

Instead of using the condition in (7.450), we use a more conservative upper limit,

$$0 < \alpha(K) < \frac{2}{\sum_{i=1}^N \lambda_i} = \frac{2}{tr[\mathbf{S}_x]} = \frac{2}{E[\|\mathbf{X}(K)\|^2]} = \frac{2}{E\left[\sum_{n=0}^{N-1} |X(n, K)|^2\right]}. \quad (7.453)$$

The motivation for this step is that  $E[\|\mathbf{X}(K)\|^2]$  can be estimated without an eigendecomposition. The denominator on the far right side of (7.453) is the expectation of the sum of the square of the received waveforms at each of the sensors at sample  $K$ . If

$$\alpha(K)\lambda_{max} \ll 2, \quad (7.454)$$

then (7.452) can be written as

$$\mathcal{M} \cong \frac{\alpha(K)}{2} \sum_{i=1}^N \lambda_i = \frac{\alpha(K)}{2} E\left[\|\mathbf{X}(K)\|^2\right]. \quad (7.455)$$

Defining the average eigenvalue as

$$\lambda_{av} = \frac{1}{N} \sum_{n=1}^N \lambda_n, \quad (7.456)$$

and assuming that  $\alpha(K)$  is constant, the transient MSE curve can be approximated with a single exponential with time constant  $\tau_{mse,av}$ ,

$$\tau_{mse,av} \simeq \frac{1}{2\alpha\lambda_{av}}, \quad (7.457)$$

and

$$\mathcal{M} \sim \frac{\alpha N \lambda_{av}}{2} = \frac{N}{4\tau_{mse,av}}. \quad (7.458)$$

Therefore, a small  $\mathcal{M}$  requires a long setting time and vice versa.

In practice, the denominator in (7.453) is approximated by an estimate obtained from the data. In most cases, we use a sample dependent  $\alpha(K)$ :

$$\alpha(K) = \frac{\gamma}{\beta + \mathbf{X}^H(K)\mathbf{X}(K)}, \quad (7.459)$$

with  $\beta > 0$  and  $0 < \gamma < 2$ , is referred to as the normalized LMS algorithm. (e.g., Goodwin and Sin [GS84] or Söderström and Stoica [SS89])

A second version,

$$\alpha(K) = \frac{\gamma}{\sigma_x^2(K)}, \quad (7.460)$$

where

$$\sigma_x^2(K) = \beta\sigma_x^2(K-1) + (1-\beta)\mathbf{X}^H(K)\mathbf{X}(K), \quad (7.461)$$

with  $0 < \beta < 1$ , is referred to as the **power normalized LMS (PNLMS)** algorithm.

Note that  $\beta$  is providing exponential weighting of  $\mathbf{X}^H(k)\mathbf{X}(k)$ , just as  $\mu$  did with the sample covariance matrix. Normally,  $\beta$  will be close to one (e.g.,  $\beta \geq 0.99$ ). The constant  $\gamma$  in the numerator satisfies  $0 < \gamma < 2$ . Typical values are  $.005 < \gamma < 0.05$ . If  $\gamma$  is too small, the step size leads to slow convergence. If  $\gamma$  is too large, the algorithm will have a large excess error or have stability problems.

All of the examples use the PNLMS algorithm.

### 7.7.2.3 Performance of the linear constrained algorithms

The statistical performance of the linear constrained LMS algorithm is analyzed in a similar manner to the unconstrained LMS algorithm. Frost [Fro72] derives a number of useful results and the reader is referred to that reference for details.

## 7.7.3 LMS Algorithm Behavior

In this section, we simulate the behavior of the LMS algorithm for various signal and interference scenarios. In Section 7.7.3.1, we study the LMS version of the MMSE beamformer. In Section 7.7.3.2, we study the LMS version of the LCMP beamformer.

### 7.7.3.1 MMSE-LMS beamformers

We consider two examples to illustrate the behavior of the Griffiths LMS algorithm in (7.416) and (7.417).

#### Example 7.7.1 (continuation, Example 7.6.4)

Consider the same model as in Example 7.6.4. The signal arrives from  $u_s = 0$  with an  $SNR = 10$  dB. A single interferer arrives from  $u_I = 0.24$  with an  $INR = 10$  dB. The trace of  $\mathbf{S}_x$  is 210 and the eigenvalues are:

$$\begin{aligned}\lambda_1 &= 117 \\ \lambda_2 &= 85 \\ \lambda_3 &= \dots = \lambda_{10} = 1.\end{aligned}$$

The average eigenvalue,  $\lambda_{av}$ , equals 21. We use the PNLMS algorithm from (7.460)–(7.461), with  $\alpha = 0.01$  and  $\beta = 0.99$ .

In Figure 7.42, we plot the average squared error and the average  $SINR_o$  versus  $K$ . We also show the steepest descent results from Example 7.6.4. In Figure 7.43, we plot the beam patterns at  $K = 20, 100, 500$ , and 1000.

We see that the PNLMS algorithm approaches the steady state result by  $K = 100$ . The beam pattern has placed a significant null on the interferer by  $K = 200$ . The single interferer case with no signal mismatch is not a challenging scenario for the LMS algorithm.

#### Example 7.7.2 (continuation, Example 7.6.5)

Consider the same model as in Example 7.6.5. The signal arrives from  $u_s = 0$  with an  $SNR = 10$  dB. The first interferer arrives from  $u_{I1} = 0.29$  with  $INR_1 = 20$  dB. The second interferer arrives from  $u_{I2} = -0.45$  with  $INR_2 = 40$  dB. The PNLMS algorithm is implemented with  $\alpha = 0.01$  and  $\beta = 0.99$ . In Figure 7.44, the average squared error versus  $K$  and the average  $SINR_o$  versus  $K$  are plotted. In Figure 7.45, representative beam patterns at  $K = 20, 100, 500$ , and 1000 are plotted.

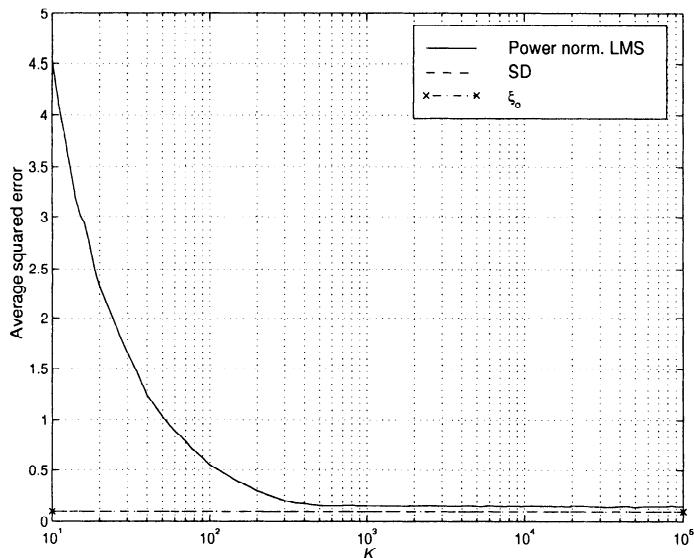
### 7.7.3.2 LCMP-LMS beamformers

In this section, we analyze the LMS implementation of the LCMP beamformer behavior for the same signal and interference model that we studied for SMI and RLS beamformers.

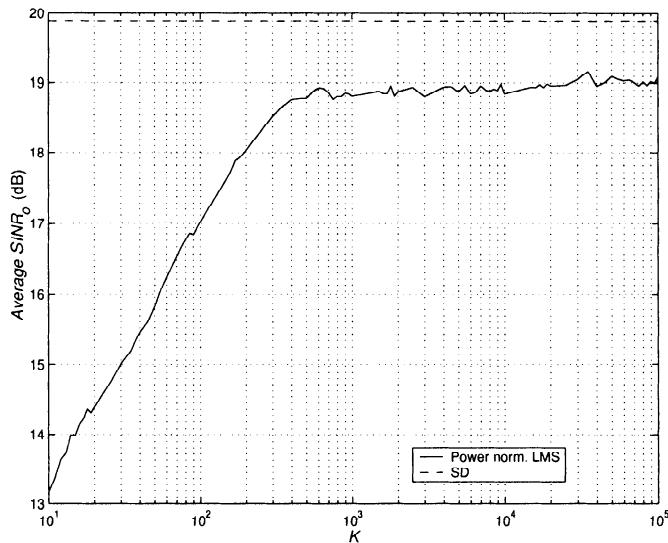
We consider a standard 10-element linear array. There is no signal mismatch or array perturbation. We do not use diagonal loading. We consider an example with two interferers in order to have two eigenvalues in the LMS algorithm.

#### Example 7.7.3

Consider a standard 10-element linear array. The desired signal arrives from  $u_s = 0$ . The interfering signals arrive from  $u_I = 0.29$  and  $0.45$ , each with an  $INR = 20$  dB and  $40$  dB, respectively. The  $SNR = 10$  dB. We implement the GSC version of the LMS MPDR beamformer. We use the PNLMS algorithm with  $\gamma = 0.01$ ,  $\beta = 0.9$  and  $\gamma = 0.01$ ,  $\beta = 0.99$ . In Figure 7.46, we plot the average  $SINR_o$  versus  $K$ . We also show the SMI with no diagonal loading result and the RLS result with  $\sigma_o^2/\sigma_w^2 = 10$  dB.



(a)



(b)

Figure 7.42 Griffiths MMSE-LMS beamformer using PNLMS algorithm:  
 $u_s = 0$ ,  $SNR = 10$  dB,  $u_I = 0.24$ ,  $INR = 10$  dB,  $\alpha = 0.01$ ,  $\beta = 0.99$ ,  
200 trials: (a) average squared error versus  $K$ ; (b) average  $SINR_o$  versus  $K$ .

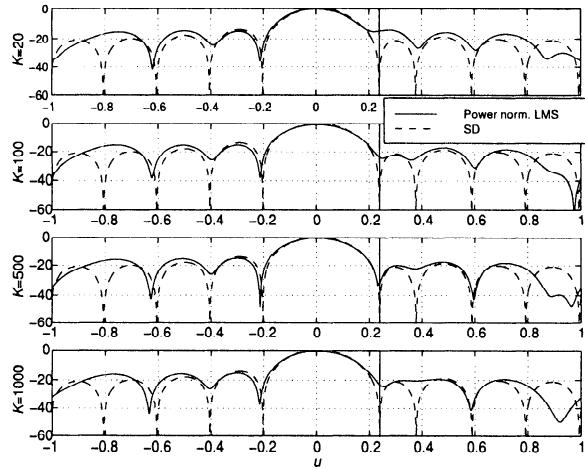


Figure 7.43 Griffiths LMS-MMSE beamformer using PNLMS algorithm:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_I = 0.24$ ,  $INR = 10$  dB,  $\alpha = 0.001$ ,  $\beta = 0.95$ ; representative beam patterns;  $K = 20, 100, 200, 500$ .

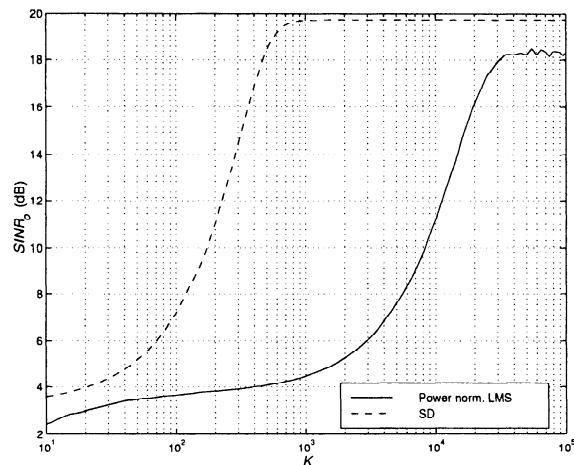


Figure 7.44 Griffiths MMSE-LMS beamformer using PNLMS algorithm:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 20$  dB,  $u_{I2} = -0.45$ ,  $INR_2 = 40$  dB,  $\alpha = 0.001$ ,  $\beta = 0.99$ ; average  $SINR_o$  versus  $K$ .

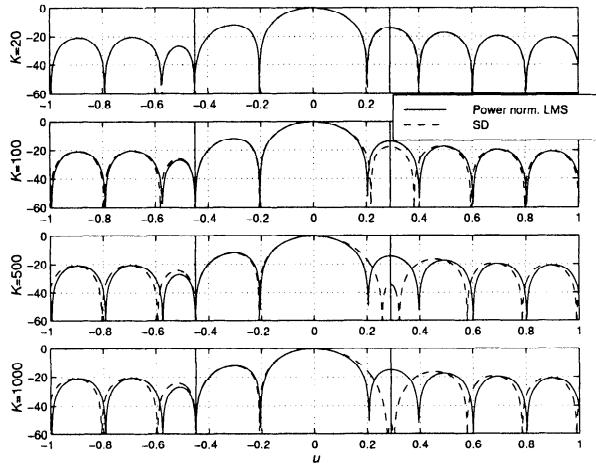


Figure 7.45 Griffiths MMSE-LMS beamformer using PNLMS algorithm:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 20$  dB,  $u_{I2} = -0.45$ ,  $INR_2 = 40$  dB,  $\alpha = 0.001$ ,  $\beta = 0.99$ ; representative beam patterns;  $K = 20, 100, 200, 500$ .

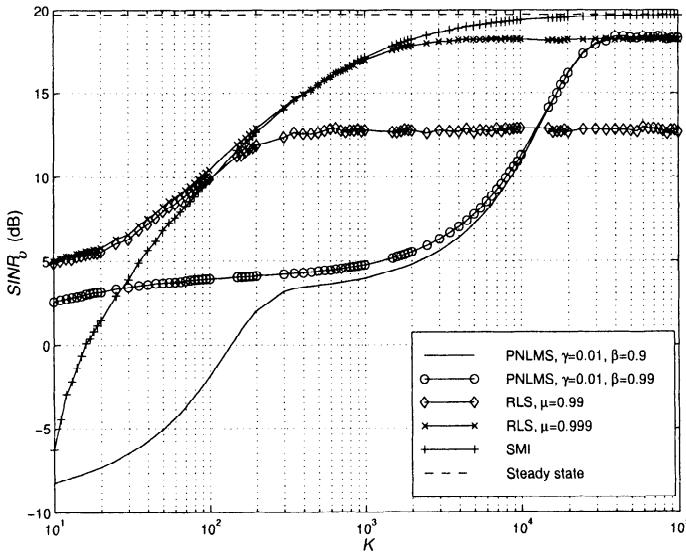


Figure 7.46 MPDR beamformer implemented using PNLMS, RLS, and SMI algorithms:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_{I1} = 0.29$ ,  $INR_1 = 10$  dB,  $u_{I2} = 0.45$ ,  $INR_2 = 40$  dB,  $\gamma = 0.01$ ,  $\beta = 0.9$  and  $0.99$ ,  $\sigma_o^2/\sigma_w^2 = 10$  dB; average  $SINR_o$  versus  $K$ .

### 7.7.4 Quadratic Constraints

In this section, we discuss quadratically constrained LMS beamformers. The discussion is parallel to the RLS discussion in Section 7.4.5. The original reference for LMS beamformers is Cox et al. [CZO87]. We develop three algorithms.

The first case utilizes fixed diagonal loading. The basic LMS-GSC algorithm is

$$\begin{aligned}\hat{\mathbf{w}}_a(K) &= \hat{\mathbf{w}}_a(K-1) + \alpha(K)\mathbf{Z}(K) \left[ Y_c^*(K) - \mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1) \right] \\ &= \hat{\mathbf{w}}_a(K-1) + \alpha(K)\hat{\mathbf{S}}_{\mathbf{z}y_c^*}(K) - \hat{\mathbf{S}}_{\mathbf{z}}(K)\hat{\mathbf{w}}_a(K-1).\end{aligned}\quad (7.462)$$

We replace

$$\hat{\mathbf{S}}_{\mathbf{z}}(K) = \mathbf{Z}(K)\mathbf{Z}^H(K) \quad (7.463)$$

with

$$\hat{\mathbf{S}}_{\mathbf{z}}(K) = \mathbf{Z}(K)\mathbf{Z}^H(K) + \sigma_L^2 \mathbf{I}. \quad (7.464)$$

Then, (7.462) becomes

$$\begin{aligned}\hat{\mathbf{w}}_a(K) &= \hat{\mathbf{w}}_a(K-1) \left( 1 - \alpha(K)\sigma_L^2 \right) + \alpha(K)\mathbf{Z}(K) \\ &\quad \left[ Y_c^*(K) - \mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1) \right].\end{aligned}\quad (7.465)$$

We refer to this algorithm as the LMS-FL (fixed loading) algorithm. It is discussed in Cox et al. [CZO87] and derived by Winkler and Schwarz [WS74] and Takao and Kikuma [TK86].

The second case is analogous to the RLS algorithm in Section 7.3.4. We update  $\hat{\mathbf{w}}_a(K)$  using

$$\tilde{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a(K-1) + \alpha(K)\mathbf{Z}(K) \left[ Y_c^*(K) - \mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1) \right]. \quad (7.466)$$

We test the norm of  $\tilde{\mathbf{w}}_a(K)$ . If <sup>22</sup>

$$\tilde{\mathbf{w}}_a^H(K)\tilde{\mathbf{w}}_a(K) \leq \gamma^2, \quad (7.467)$$

we set

$$\hat{\mathbf{w}}_a(K) = \tilde{\mathbf{w}}_a(K). \quad (7.468)$$

If

$$\tilde{\mathbf{w}}_a^H(K)\tilde{\mathbf{w}}_a(K) > \gamma^2, \quad (7.469)$$

---

<sup>22</sup>We use  $\gamma^2$  instead of  $\alpha^2$  to avoid confusion with the LMS step size parameter  $\alpha(K)$ .

we add a diagonal loading term to  $\tilde{\mathbf{w}}_a(K)$ ,

$$\begin{aligned}\hat{\mathbf{w}}_a(K) &= \tilde{\mathbf{w}}_a(K) - \alpha(K)\beta(K)\hat{\mathbf{w}}_a(K-1) \\ &= \tilde{\mathbf{w}}_a(K) - \beta(K)\hat{\mathbf{v}}_{lms}(K).\end{aligned}\quad (7.470)$$

where

$$\hat{\mathbf{v}}_{lms}(K) = \alpha(K)\hat{\mathbf{w}}_a(K-1). \quad (7.471)$$

We then solve the quadratic equation

$$\begin{aligned}&\hat{\mathbf{v}}_{lms}^H(K)\hat{\mathbf{v}}_{lms}(K)\beta^2(K) \\ &-2Re\left[\hat{\mathbf{v}}_{lms}^H(K)\tilde{\mathbf{w}}_a(K)\right]\beta(K) \\ &+\tilde{\mathbf{w}}_a^H(K)\tilde{\mathbf{w}}_a(K) - \gamma^2 = 0.\end{aligned}\quad (7.472)$$

We solve (7.472) and use the results in the following manner. The two roots of (7.472) are denoted by  $\beta_1(K)$  and  $\beta_2(K)$ .

If there are two positive real roots, we use the smallest root. In this case we meet the constraint. If there are two negative real roots, we set  $\beta_1(K)$  and  $\beta_2(K) = 0$  and do not meet the constraint. Simulations indicate that this result happens infrequently. If there are two complex roots, they have the same real part. We use the real part as the diagonal load. We refer this algorithm as the LMS-VL (variable loading) algorithm.

The third case is the LMS-SP (scaled projection) algorithm due to Cox et al. [CZ087].

Once again, we require

$$\|\mathbf{w}_a\|^2 \leq \gamma^2. \quad (7.473)$$

We update  $\hat{\mathbf{w}}_a(K)$  recursively using (7.466),

$$\begin{aligned}\tilde{\mathbf{w}}_a(K) &= \hat{\mathbf{w}}_a(K-1) + \alpha(K)\mathbf{Z}(K)\left[Y_c^*(K) - \mathbf{Z}^H(K)\hat{\mathbf{w}}_a(K-1)\right] \\ &= \hat{\mathbf{w}}_a(K-1) + \alpha(K)\mathbf{Z}(K)e_p^*(K) \\ &= \hat{\mathbf{w}}_a(K-1) + \mathbf{g}_z(K)e_p^*(K).\end{aligned}\quad (7.474)$$

Note that  $\hat{\mathbf{w}}_a^H(K-1)$  satisfies the constraint in (7.473). The next step is shown graphically in Figure 7.47.

If  $\tilde{\mathbf{w}}_a(K)$  satisfies (7.473), we use it as  $\hat{\mathbf{w}}_a(K)$ . Otherwise we scale it to satisfy (7.473). Thus,

$$\hat{\mathbf{w}}_a(K) = \begin{cases} \tilde{\mathbf{w}}_a(K), & \|\tilde{\mathbf{w}}_a(K)\|^2 \leq \gamma^2, \\ c(K)\tilde{\mathbf{w}}_a(K), & \|\tilde{\mathbf{w}}_a(K)\|^2 > \gamma^2, \end{cases} \quad (7.475)$$

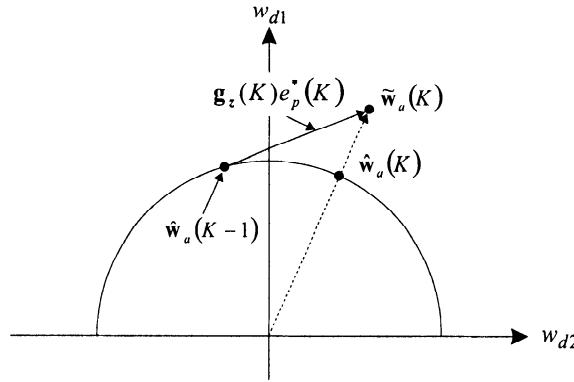


Figure 7.47 Scaling the tentative update vector.

where  $c(K)$  scales  $\tilde{\mathbf{w}}_a(K)$  so that  $\hat{\mathbf{w}}_a(K)$  satisfies (7.473),

$$c(K) = \frac{\gamma}{\|\tilde{\mathbf{w}}_a(K)\|}. \quad (7.476)$$

We see that all of the constraints are satisfied at each step in the recursion.

We consider an example to show the performance of the three diagonal loading algorithms.

#### Example 7.7.4

Consider a standard 10-element array. The nominal signal direction is  $u_s = 0$ . We implement an MPDR beamformer using LMS-FL, LMS-VL, and LMS-SP. There is a single interferer at  $u_I = 0.29$  with an  $INR = 20$  dB.

The signal arrives from  $u_a = 0.03$  with a  $SNR = 10$  dB. We use  $T_o = 0.2$  in the LMS-VL and LMS-SP algorithms. We use  $\sigma_L^2/\sigma_w^2 = 10$  dB in the LMS-FL algorithm.

In Figure 7.48, we plot the average  $SINR_o$  versus  $K$ . The  $\overline{SINR_o}$  behavior of the three algorithms is identical. Note that the steady state value is 20 dB, so the LMS is converging slowly.

In Figure 7.49, we plot representative beam patterns. The three algorithms are similar. A significant null does not appear until  $K = 1,000$ . Due to the slow convergence and diagonal loading there is no signal nulling.

The general conclusion is:

- (i) The LMS-VL and LMS-SP behave in an almost identical manner under most scenarios. In most applications, the LMS-SP algorithm will be used.
- (ii) The LMS algorithm convergence will be slow in many cases of interest.

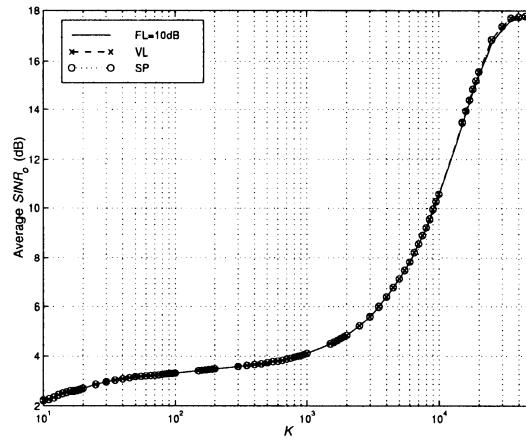


Figure 7.48 MPDR-LMS beamformer with signal mismatch using fixed loading, variable loading, and scaled projection:  $u_s = 0, u_a = 0.03, SNR = 10$  dB,  $u_I = 0.29, INR = 20$  dB,  $T_o = 0.2, \sigma_L^2/\sigma_w^2 = 10$  dB; average  $SINR_o$  versus  $K$ .

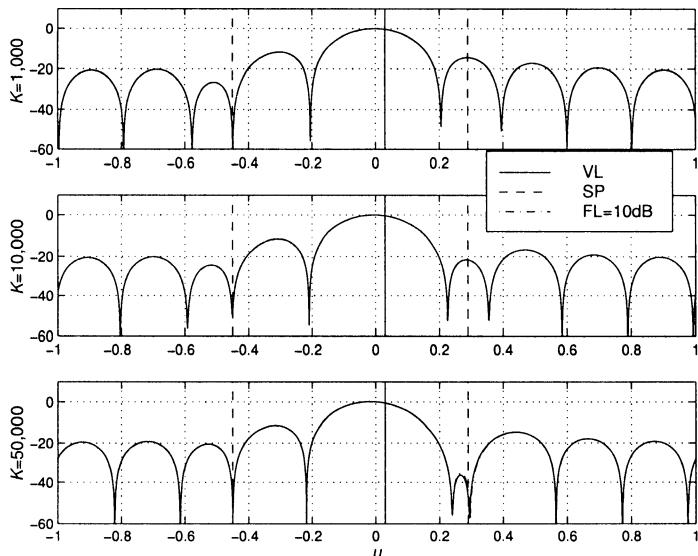


Figure 7.49 MPDR-LMS beamformer with signal mismatch using fixed loading, variable loading, and scaled projection:  $u_s = 0, u_a = 0.03, SNR = 10$  dB,  $u_I = 0.29, INR = 20$  dB,  $T_o = 0.2, \sigma_L^2/\sigma_w^2 = 10$  dB; representative beam patterns at  $K = 10, 100$ , and  $1,000$ .

### 7.7.5 Summary: LMS algorithms

In this section, we discuss a parallel approach to adaptive arrays and then summarize our results.

#### 7.7.5.1 Howells-Applebaum Algorithms

In this section, we have discussed various LMS algorithms. They represent one of the paths along which adaptive arrays have evolved. The other path originated with the work of Howells and Applebaum at General Electric in the early 1960s. The first practical electronic steering of an antenna null in the direction of a jammer was invented by Paul Howells at General Electric in Syracuse, NY. His 1959 U.S. Patent, number 3,202,990, was titled “Intermediate Frequency Side-Lobe Cancellor.” Their original work was published as a Syracuse University Research Corporation report [App66]. Because the circulation was limited, [App66] was republished as [App76]. Several subsequent papers (e.g., [AC76]) discuss the original work and extensions.

In their original work, Howells and Applebaum considered a narrowband array and used the criterion of maximum *SNR*. They also introduced the idea of quiescent pattern control. They implemented the resulting adaptive processor with a closed-loop analog system. The feedback path contained an integrator that they approximated with a low pass-filter. The resulting system is the analog version of the Griffiths LMS beamformer with DL.

Additional historical information is given in Tufts [Tuf98].

#### 7.7.5.2 Summary

In Section 7.7, we have developed LMS algorithms for implementing adaptive beamformers. Although they are characterized by slow convergence in many situations, they are widely used in practice because of their computational simplicity.

Our discussions of the LMS algorithm has been reasonably brief because a number of comprehensive discussions are available in the literature.

For further discussion of LMS algorithms for adaptive beamforming and adaptive filtering, the reader is referred to Widrow and Stearns [WS85], Haykin [Hay96], Compton [Com88], Monzingo and Miller [MM80], and Glen-tis et al. [GBT99].

## 7.8 Detection of Signal Subspace Dimension

In Section 6.8, we derived several eigenvector beamformers and saw that they offered an improvement in performance when the statistics are known. In practice, we must estimate the eigenvalues and eigenvectors using the incoming data and then select an appropriate subspace. In the case in which the desired signals and interferers are plane-wave signals we can refer to this subspace as the signal-plus-interference subspace. For simplicity, we consider all of the plane-wave signals to be signals of interest and refer to this subspace as the signal subspace.

The eigenvector beamformers provide motivation for our current discussion of detecting the subspace dimension. However, when we study parameter estimation in Chapters 8 and 9, we find that almost all of the estimation algorithms assume that the number of signals are known. Therefore, we develop the detection problem in a general framework, so that the results are useful in both the adaptive beamforming problem and the parameter estimation problem. We apply the results to the adaptive eigenspace beamformer in Section 7.9.

The techniques that we develop in this section are non-parametric techniques. Although we model the received vector as

$$\mathbf{X}(k) = \mathbf{V}(\psi)\mathbf{F}(k) + \mathbf{W}(k), \quad k = 1, 2, \dots, K, \quad (7.477)$$

we do not exploit the structure of  $\mathbf{V}(\psi)$  in our algorithms. In (7.477),  $\mathbf{F}(k)$  is a  $d \times 1$  source vector that is a complex Gaussian vector.  $\mathbf{V}(\psi)$  is an  $N \times d$  array manifold matrix and  $\mathbf{W}(k)$  is a complex Gaussian vector with a diagonal spectral matrix ( $\sigma_w^2 \mathbf{I}$ ). The number of signals  $d$  is unknown and we want to estimate its value. Because  $d$  is an integer, we refer to the problem as a detection problem.

If none of the signals are coherent with each other, then the rank of the subspace equals the number of signals.

However, we recall from (5.252) that if

$$|\rho_{ij}| = 1, \quad (7.478)$$

for some value of  $ij$ , then we will have one or more zero eigenvalues and

$$d' < d. \quad (7.479)$$

As  $|\rho_{ij}|$  approaches one, the algorithms have a harder time correctly estimating the number of signals.

All of the approaches in this section perform an eigendecomposition of the sample spectral matrix or a singular value decomposition of the data matrix to find a set of estimated eigenvalues. The algorithms test some function of these eigenvalues to determine  $d$  or  $d'$ .

In Section 7.8.1, we develop the set of detection algorithms that are most widely used in the array processing. All of the tests in Section 7.8.1 utilize some function of the eigenvalues of  $\hat{\mathbf{S}}_x$  as the data component of the test. The performance of the various algorithms is analyzed for a sequence of examples.

In Section 7.8.2, we develop a detection algorithm that utilizes the eigenvectors of  $\hat{\mathbf{S}}_x$ .

### 7.8.1 Detection Algorithms

We use the familiar frequency-domain snapshot model in (7.477). We assume the columns of  $\mathbf{V}$  are linearly independent.

The spectral matrix is

$$\mathbf{S}_x = \mathbf{V} \mathbf{S}_f \mathbf{V}^H + \sigma_w^2 \mathbf{I}. \quad (7.480)$$

We estimate  $d$ , the rank of  $\mathbf{S}_f$ .

If  $\mathbf{V} \mathbf{S}_f \mathbf{V}^H$  is of rank  $d$ , the eigendecomposition of  $\mathbf{S}_x$  can be written as

$$\mathbf{S}_x = \mathbf{U}_S \Lambda_S \mathbf{U}_S^H + \mathbf{U}_N \Lambda_N \mathbf{U}_N^H, \quad (7.481)$$

where

$$\Lambda_N = \text{diag} \left[ \begin{array}{ccc} \sigma_w^2 & \cdots & \sigma_w^2 \end{array} \right], \quad (7.482)$$

is a diagonal matrix with  $N - d$  elements.

In practice, we have  $\hat{\mathbf{S}}_x$  available and we compute

$$\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \hat{\lambda}_3 \cdots \hat{\lambda}_d \geq \hat{\lambda}_{d+1} \geq \cdots \hat{\lambda}_N, \quad (7.483)$$

via an eigendecomposition of  $\hat{\mathbf{S}}_x$  or an SVD of the data matrix. We want to detect the value of  $d$ .

The first category of tests are referred to as sequential hypothesis (SH) tests or sphericity tests. They originated in the statistical field (e.g., [Mau40], [And63], or [Mui82]).

We want to find the likelihood ratio between the hypothesis that the  $(N - d)$  smallest eigenvalues are equal versus the hypothesis that only the  $(N - d - 1)$  smallest eigenvalues are equal. We use  $d$  as the variable in our test.

Anderson [And63] showed that, for  $K \gg N$ , if  $\mathbf{V}$  is of rank  $d$ , the  $N - d$  smallest eigenvalues cluster around  $\sigma_w^2$ . Specifically,  $\hat{\lambda}_k - \sigma_w^2 = O\left(K^{-\frac{1}{2}}\right)$  for  $k = d + 1, \dots, N$ .

Anderson [And63] also showed that a sufficient statistic is

$$L_d(d) = K(N-d) \ln \left\{ \frac{\frac{1}{N-d} \sum_{k=d+1}^N \hat{\lambda}_k}{\left( \prod_{k=d+1}^N \hat{\lambda}_k \right)^{\frac{1}{N-d}}} \right\}. \quad (7.484)$$

The term in braces is the ratio of the arithmetic mean of the  $(N-d)$  smallest eigenvalues to the geometric mean of the  $(N-d)$  smallest eigenvalues. We now have a 1-D optimization problem to find  $\hat{d}$ . Note that, if the  $N-d$  smallest eigenvalues are equal, then  $L_d(d) = 0$ .

One can show that asymptotically  $(2L(d))$  corresponds to a chi-squared random variable  $\chi^2((N-d)^2 - 1)$  (e.g., Gupta [Gup65]). Thus, if

$$\nu = 2L_d(d), \quad (7.485)$$

then

$$f_\nu(d) = \frac{\nu^{\left(\frac{1}{2}\left((N-d)^2-1\right)-2\right)} \cdot e^{\left(-\frac{1}{2}\nu\right)}}{2^{\left(\frac{(N-d)^2-1}{2}\right)} \cdot \Gamma\left(\frac{(N-d)^2-1}{2}\right)}. \quad (7.486)$$

We choose an arbitrary confidence interval (e.g., 99%). We assume  $H_0$  (i.e.,  $d = 0$ ) is true and compute  $\gamma_{99}^{(0)}\left(\chi^2_{N^2-1}\right)$ . If

$$\nu(0) \leq \gamma_{99}^{(0)}, \quad (7.487)$$

we set  $\hat{d} = 0$ . If  $\nu(0) > \gamma_{99}^{(0)}$ , we compute  $\gamma_{99}^{(1)}\left(\chi^2_{(N-1)^2-1}\right)$ . Then, if

$$\nu(1) \leq \gamma_{99}^{(1)}, \quad (7.488)$$

we set  $\hat{d} = 1$ . If  $\nu(1) > \gamma_{99}^{(1)}$ , we continue the process until we find

$$\nu(d) \leq \gamma_{99}^{(d)}\left(\chi^2_{(N-d)^2-1}\right). \quad (7.489)$$

Bartlett [Bar54] and Lawley [Law56] developed tests based on (7.484). Simkins [Sim80] applied these results to the direction-of-arrival estimation problem.

One of the problems with the above approaches is the choice of the threshold. In addition, a sequence of tests is required. We do not pursue

sequential hypothesis tests further in the text. The reader is referred to the references and the problems for representative results.

The second category of tests utilize  $L_d(d)$  and add a penalty function related to the degrees of freedom. The result is a function of  $d$  that we then minimize. The two resulting tests were obtained using different approaches.

Akaike [Aka74] introduced an information-theoretic criterion, which is referred to in the literature as the Akaike Information Criterion (AIC). He considered a parameterized family of models with probability densities  $p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})$ . He proposed to choose the model that has the minimum AIC where

$$AIC = -2 \ln p_{\mathbf{x}}(\mathbf{x}|\hat{\boldsymbol{\theta}}) + 2k_p, \quad (7.490)$$

where  $\hat{\boldsymbol{\theta}}$  is the ML estimate of  $\boldsymbol{\theta}$  and  $k_p$  is the number of freely adjusted parameters in  $\boldsymbol{\theta}$ .<sup>23</sup> The first term is the log-likelihood for the ML estimate of  $\boldsymbol{\theta}$ . The second term is a correction term. Akaike introduced it so that the AIC is an unbiased estimate of Kullback-Liebler distance between  $p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})$  and  $p_{\mathbf{x}}(\mathbf{x}|\hat{\boldsymbol{\theta}})$ .

Two different approaches were taken by Schwartz [Sch78] and Rissanen [Ris78]. Schwartz [Sch78] utilized a Bayesian approach, assigning a prior probability to each model, and selected the model with the largest *a posteriori* probability. Rissanen used an information theoretic argument. One can think of the model as an encoding of the observation. He proposed choosing the model that gave the minimum code length. In the large sample limit, both approaches lead to the same criterion

$$MDL = -\ln p_{\mathbf{x}}(\mathbf{x}|\hat{\boldsymbol{\theta}}) + \frac{1}{2} k_p \ln K, \quad (7.491)$$

where MDL denotes minimum description length.

In [WK85] the AIC and MDL criteria were applied to the array problem.<sup>24</sup> The starting point is to define a family of spectral matrices,

$$\mathbf{S}_{\mathbf{x}}^{(d)} = \boldsymbol{\Psi}^{(d)} + \sigma_w^2 \mathbf{I}, \quad d = 0, 1, \dots, N-1, \quad (7.492)$$

where  $\boldsymbol{\Psi}^{(d)}$  is a non-negative definite matrix of rank  $d$  and  $\sigma_w^2$  is an unknown scalar. We write  $\mathbf{S}_{\mathbf{x}}^{(d)}$  as

$$\mathbf{S}_{\mathbf{x}}^{(d)} = \sum_{i=1}^d \lambda_i \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^H + \sigma_w^2 \sum_{i=d+1}^N \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^H. \quad (7.493)$$

---

<sup>23</sup> $\boldsymbol{\theta}$  will be defined in the context of the array problem in (7.494).

<sup>24</sup>Our discussion in the next part of this section follows Wax and Kailath [WK85].

The parameter vector  $\boldsymbol{\theta}^{(d)}$  of the model is

$$\boldsymbol{\theta}^{(d)} = \left[ \lambda_1, \dots, \lambda_d, \sigma_w^2, \Phi_1^T, \dots, \Phi_d^T \right]^T. \quad (7.494)$$

Note that  $\boldsymbol{\theta}$  does not include  $\psi$ . Also observe that we are estimating the parameters in the eigendecomposition of  $\mathbf{S}_x^{(d)}$  without imposing any structure on it.

The ln likelihood function was derived in (7.7),

$$L\left(\boldsymbol{\theta}^{(d)}\right) = -K \ln \det \mathbf{S}_x^{(d)} - K \operatorname{tr} \left[ \left[ \mathbf{S}_x^{(d)} \right]^{-1} \mathbf{C}_x \right] \quad d = 0, 1, \dots, N-1, \quad (7.495)$$

where  $\mathbf{C}_x$  is the sample spectral matrix. From (7.9), the maximum likelihood estimate of  $\mathbf{S}_x^{(d)}$  is

$$\hat{\mathbf{S}}_x^{(d)} = \mathbf{C}_x, \quad d = 0, 1, \dots, N-1. \quad (7.496)$$

Now expand  $\mathbf{C}_x$  using an eigendecomposition,

$$\mathbf{C}_x = \sum_{i=1}^N \hat{\lambda}_i \hat{\Phi}_i \hat{\Phi}_i^H = \sum_{i=1}^d \hat{\lambda}_i \hat{\Phi}_i \hat{\Phi}_i^H + \sum_{i=d+1}^N \hat{\lambda}_i \hat{\Phi}_i \hat{\Phi}_i^H, \quad (7.497)$$

where  $\hat{\lambda}_i$  and  $\hat{\Phi}_i$  are the eigenvalues and eigenvectors of the sample covariance matrix  $\mathbf{C}_x$ . Substituting (7.493) and (7.497) into (7.496) gives the maximum likelihood estimate of the components of  $\boldsymbol{\theta}^{(d)}$ :

$$(\hat{\lambda}_i)_{ml} = \hat{\lambda}_i, \quad i = 1, \dots, d, \quad (7.498)$$

$$(\hat{\Phi}_i)_{ml} = \hat{\Phi}_i, \quad i = 1, \dots, d, \quad (7.499)$$

$$(\hat{\sigma}_w^2)_{ml} = \frac{1}{N-d} \sum_{i=d+1}^N \hat{\lambda}_i. \quad (7.500)$$

Thus, the maximum likelihood estimate

$$\hat{\boldsymbol{\theta}}^{(d)} = \left[ \hat{\lambda}_1, \dots, \hat{\lambda}_d, \hat{\sigma}_w^2, \hat{\Phi}_1^T, \dots, \hat{\Phi}_d^T \right]^T, \quad (7.501)$$

where the  $ml$  subscript is dropped for simplicity. Substituting (7.501) and (7.500) in (7.495) gives

$$L_d(d) = K(N-d) \ln \left\{ \frac{\frac{1}{N-d} \sum_{i=d+1}^N \hat{\lambda}_i}{\left( \prod_{i=d+1}^N \hat{\lambda}_i \right)^{\frac{1}{N-d}}} \right\}, \quad (7.502)$$

where we have dropped terms that do not depend on  $d$ . This is the same expression as Anderson's sufficient statistic in (7.484).

It is convenient to write the AIC test as

$$L_d(d) + p(d), \quad (7.503)$$

where  $p(d)$  is a penalty function. The penalty function is determined by the number of degrees of freedom in the model (the number of free parameters in  $\theta^{(d)}$ ).

The parameter vector  $\theta^{(d)}$  consists of  $d+1$  eigenvalues and  $d$  eigenvectors.<sup>25</sup> The eigenvalues are real and hence count as  $d+1$  real parameters, whereas the eigenvectors are complex, of unit norm and mutually orthogonal. To count the number of degrees-of-freedom (DOF) required to describe the  $d$  eigenvectors, observe that because the eigendecomposition of a complex matrix is invariant to multiplication of each eigenvector by a pure phase factor, we can describe an eigenvector with  $2N-1$  real parameters, say by fixing the first element to be real. Since the eigenvectors obey certain constraints, we have to deduct  $d$  DOF due to their unit norm and  $2(1/2(d(d-1)))$  due to their mutual orthogonalization. Therefore, the total number of degrees of freedom is

$$\begin{aligned} k_p &= d+1 + 2Nd - 2d - d(d-1) \\ &= d(2N-d) + 1. \end{aligned} \quad (7.504)$$

We can drop the one, so

$$p(d) = d(2N-d), \quad (7.505)$$

and the AIC test is

$$AIC(d) \triangleq \{L_d(d) + [d(2N-d)]\}, \quad (7.506)$$

and

$$\hat{d}_{AIC} = \arg \min_d \{AIC(d)\}. \quad (7.507)$$

For the MDL test,

$$\begin{aligned} MDL(d) &= L_d(d) + \frac{1}{2}p(d) \ln K \\ &= L_d(d) + \frac{1}{2}[d(2N-d) + 1] \ln K, \end{aligned} \quad (7.508)$$

---

<sup>25</sup>This explanation is due to M. Wax (private communication).

and

$$\hat{d}_{MDL} = \arg \min_d \{ MDL(d) \}. \quad (7.509)$$

In [WK85], it is shown that the MDL estimate is consistent.<sup>26</sup> In other words, as  $K$  goes to infinity,  $\hat{d}_{MDL}$  approaches  $d$ . It is also shown that the AIC is inconsistent and, asymptotically, tends to overestimate the number of signals. However, we will find that, for a small  $K$ , the AIC generally has a higher probability of a correct decision.

We have seen the benefit of FB averaging in a coherent or correlated signal environment. We now discuss how the above detection tests must be modified to account for the use of  $\mathbf{C}_{\mathbf{x},fb}$  instead of  $\mathbf{C}_{\mathbf{x}}$ , the forward-only sample spectral matrix.

The problem is solved by Xu et al. [XRK94]. They derive sequential hypothesis (SH), MDL, and AIC tests that are closely related to the above tests. Their key result is that the tests have the same structure; the only change is in the penalty function.

The FB-MDL and FB-AIC functions are

$$AIC_{FB}(d) = L_d(d) + \frac{1}{2}d(2N - d + 1), \quad (7.510)$$

and

$$MDL_{FB}(d) = L_d(d) + \frac{1}{4}d(2N - d + 1) \ln K. \quad (7.511)$$

where  $L_d(d)$  is given by (7.502) where the  $\hat{\lambda}_i$  are the eigenvalues of  $\mathbf{C}_{\mathbf{x},fb}$ . The effect of the FB averaging is to reduce the free adjustable parameters by a factor of 2. We find the value of  $d$  that minimizes (7.510) or (7.511) and denote it by  $\hat{d}$ . Xu et al. [XRK94] also shows that FB-MDL is consistent.

Wong et al. [WZRY90] derive an alternative approach to detecting the number of signals. They argue that the number of signals is independent of the orientation of the array, so the eigenvectors provide little information. They derive a modified likelihood function consisting of the marginal probability density function of the eigenvalues. They derive two modified criteria using this modified likelihood function and the penalty functions of AIC and MDL. One of the criteria offers improved performance in a number of scenarios and should be considered as a candidate solution. Wu and Tam ([TW96], [WT01]) also discuss this approach. The reader is referred to the above references for details.

Another category of tests is due to Tufts and his colleagues (e.g., Shah and Tufts [ST94]) and focuses on scenarios with short data records. The

---

<sup>26</sup>See subsequent discussion by Zhao et al. ([ZKB87], [ZKB86]).

method consists of sequential constant false alarm rate tests on the sums of the eigenvalues of  $\hat{\mathbf{S}}_x$ . In several scenarios, it offers improved performance compared to AIC and MDL. The reader is referred to the above reference for details. An approach that deals with unknown correlated noise fields is derived in Stoica and Cedervall [SC97].

As pointed out in the introduction to this section, all of the algorithms in this section are nonparametric. In Chapter 8, we revisit the detection problem and discuss parametric detection algorithms.

We now consider a sequence of examples in which we simulate the performance of AIC and MDL algorithms.

The examples are chosen to illustrate how the following factors affect the performance:

- (i) Number of signals;
- (ii) Signal strengths;
- (iii) Signal separations;
- (iv) Signal correlation;
- (v) Number of snapshots.

There are three possible outcomes of the tests:<sup>27</sup>

- (i)  $\hat{d} = d$ , which is the desired result. We denote the probability that  $\hat{d} = d$  as  $P_D$  (the probability of correct detection).
- (ii)  $\hat{d} < d$ . In this case, we have underestimated the signal subspace. We denote the probability as  $P_M$  (probability of miss). In almost all applications, we want to avoid this event because it causes us to lose one or more signals.
- (iii)  $\hat{d} > d$ . In this case, we have overestimated the signal subspace. We denote the probability as  $P_F$  (probability of false alarm). In most applications this event causes some degradation in performance. However, the extra eigenvector(s) is due to noise. If we try to track the eigenspace as additional data arrive, the subspace tracker drops it and reduces the dimension. If we do block processing, the next block of data will put the extra eigenvector(s) in a different place (or omit them). We rely on this subsequent processing to correct overestimation.

---

<sup>27</sup>Our comments on the outcomes are in the context of beamforming. We discuss the effects on parameter estimation algorithms later.

We now consider a sequence of examples to explore the behavior. In all of the examples, we consider a standard 10-element linear array.

In the first four examples, we consider a standard 10-element linear array with two equal-power signals impinging on it. The two-signal case has been studied in detail by Kaveh and Wang (see Chapter 5 of [Hay91]). They derive analytic expressions for  $P_M$  and  $P_F$  based on the asymptotic properties of the eigenvalues and also provide simulation results. We use the simulation approach.

In this case, the outcomes are:

$$P_M : \hat{d} = 0 \text{ or } 1$$

$$P_D : \hat{d} = 2$$

$$P_F : \hat{d} \geq 3$$

We consider four algorithms: AIC, MDL, AIC-FB, and MDL-FB.

#### **Example 7.8.1**

In this example, the signal DOAs are symmetric about  $u = 0$  and the signal separation corresponds to 0.5 HPBW. The signals are uncorrelated. We assume  $K = 100$ . We vary the  $ASNR$ .

In Figure 7.50, we plot  $P_D$  versus  $ASNR^{28}$  for the four algorithms. In Figure 7.51, we plot  $P_M$  versus  $ASNR$  for the four algorithms. In Figure 7.52, we plot  $P_F$  versus  $ASNR$  for the four algorithms.

We see that the AIC algorithm provides correct detection with an  $ASNR$  that is 3–4 dB lower than the MDL algorithms in the  $0.2 \leq P_D \leq 0.9$  range. However as the  $ASNR$  increases, the  $P_D$  levels off at about  $P_D = 0.92$  and the  $P_{FA}$  is about 0.08. Above  $ASNR = 6$  dB,  $P_M$  for AIC is zero. The AIC-FB has a higher  $P_D$  for low  $ASNR$ , but levels off at about 0.83 as the  $ASNR$  increases.

The  $P_D$  for the MDL algorithm is 1.0 for  $ASNR \geq 9$  dB (7 dB for MDL-FB).

These results suggest that, for this particular scenario, if we can be certain that the  $ASNR$  will be greater than 10 dB for the signals of interest, then we would use MDL or MDL-FB. However, if we are interested in signals whose  $ASNR$  may be as low as 0 dB, then we use AIC or AIC-FB and try to eliminate false alarms by subsequent processing.

#### **Example 7.8.2** (continuation, Example 7.8.1)

Consider the same model as in Example 7.8.1 except  $K$ , the number of snapshots is allowed to vary. The signal separation is 0.5 HPBW.

We plot the required  $ASNR$  versus  $K$  to achieve  $P_D = 0.2$ , 0.5, and 0.8. In Figure 7.53, we consider the AIC algorithm. In Figure 7.54, we consider the MDL algorithm.

---

<sup>28</sup>The  $ASNR$  is defined as  $N(SNR)$ . It is useful to plot results versus  $ASNR$ , because, in most cases, for  $N \geq 10$ , the result will not depend on  $N$ .

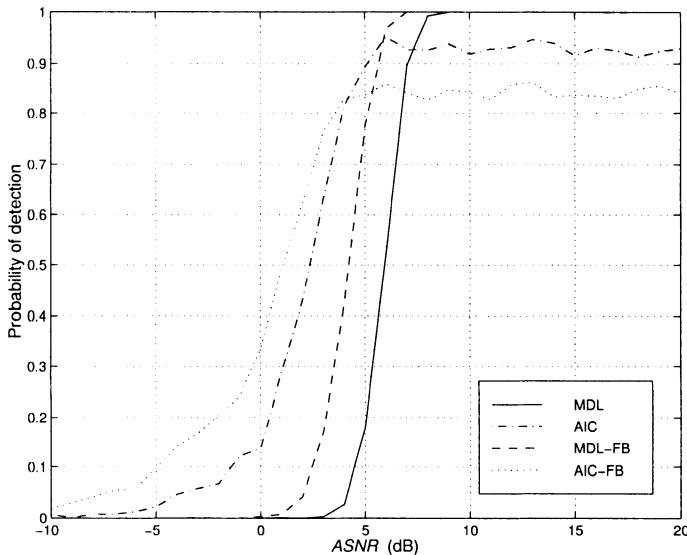


Figure 7.50 Performance of AIC and MDL algorithms versus  $ASNR$ : two equal-power signals,  $\Delta u = 0.5$  HPBW,  $K = 100$ , 1,000 trials,  $P_D$  versus  $ASNR$ .

We see that the AIC curves are parallel with a decrease in the detection threshold of about 5 dB/decade. The MDL curves are parallel with a similar slope.

**Example 7.8.3** (continuation, Example 7.8.1)

Consider a standard 10-element array. The signals have equal power and are uncorrelated. The signals impinge on the array from  $\pm\Delta u/2$ .

In Figure 7.55, we plot the  $SNR$  required to achieve  $P_D = 0.2$ , 0.5, and 0.8 versus  $\Delta u/\text{HPBW}$  for the AIC-FO algorithm. In Figure 7.56, we plot the same results for the MDL-FO algorithm.

We see that the threshold decreases at a rate of about 20 dB/decade. The MDL curves have a similar slope, but are about 4 dB higher.

**Example 7.8.4** (continuation, Example 7.8.1)

In this example, we study the behavior as the number of signals,  $d$ , is varied. Consider a standard 10-element linear array. We assume that there are  $d$  signals placed symmetrically about  $u = 0$ . We consider two spacings. In case 1, the spacing between the signals is 0.5 HPBW. In case 2, the spacing between the signals is 1.0 HPBW. The number of snapshots is 100. The signals are equal power and uncorrelated.

In Figure 7.57, the signal separation is 0.5 HPBW. We plot  $P_D$  versus the  $ASNR$  for each signal for the AIC-FO and MDL-FO algorithms. If we examine the  $ASNR$  required for  $P_D = 0.5$ , we see that the additional  $ASNR$  required is about  $(d + 4)$  dB for each additional signal up to  $d = 7$ , and then it is about 11 dB for  $d = 8$  and 9.

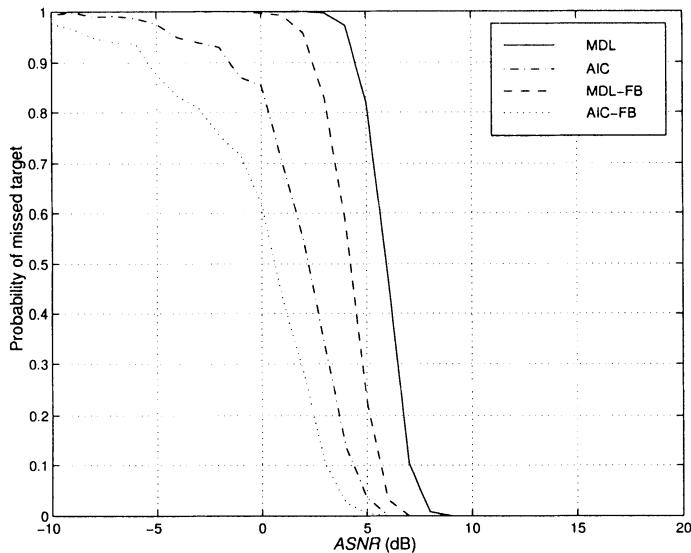


Figure 7.51 Performance of AIC and MDL algorithms versus  $ASNR$ : two equal-power signals,  $\Delta u = 0.5$  HPBW,  $K = 100$ , 1,000 trials,  $P_M$  versus  $ASNR$ .

In Figure 7.58, the signal separation is 1.0 HPBW. We plot the same results as in Figure 7.57. We see that the effect of increasing  $d$  is significantly less.

In this subsection, we have studied AIC and MDL tests for detecting the number of plane-wave signals. We first considered the familiar case of two equal-power uncorrelated signals separated by 0.5 HPBW and  $K = 100$ . We then considered the effects of:

- (i)  $K$ , the number of snapshots;
- (ii)  $\Delta u$ , the signal separation;
- (iii)  $d$ , the number of signals.

In all of these cases, the AIC algorithm performed better than the MDL algorithm in the  $0.2 \leq P_D \leq 0.9$  range. However, as the  $ASNR$  (or  $K$ ) increased, the  $P_D$  of the AIC algorithm did not approach unity and the AIC algorithm overestimated the number of signals. The  $P_D$  of the MDL algorithm approached unity as the  $ASNR$  (or  $K$ ) increased.

The choice of the algorithm will depend on the anticipated scenario and the subsequent signal processor that uses the estimate of  $d$ . In Section

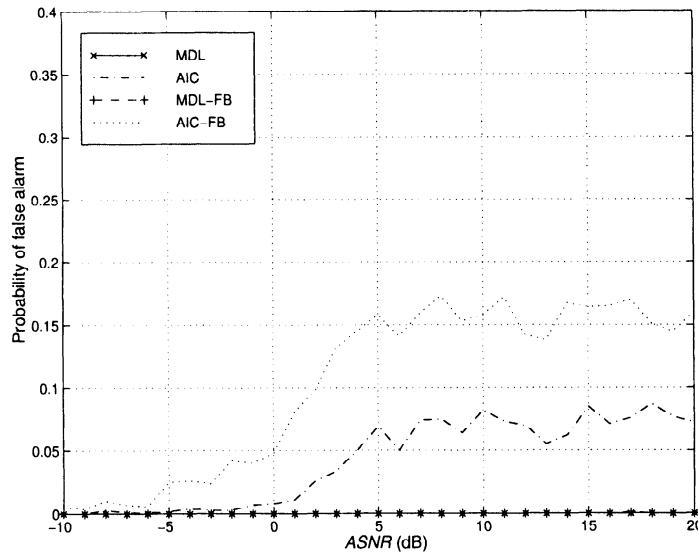


Figure 7.52 Performance of AIC and MDL algorithms versus  $ASNR$ : two equal-power signals,  $\Delta u = 0.5$  HPBW,  $K = 100$ , 1,000 trials,  $P_F$  versus  $ASNR$ .

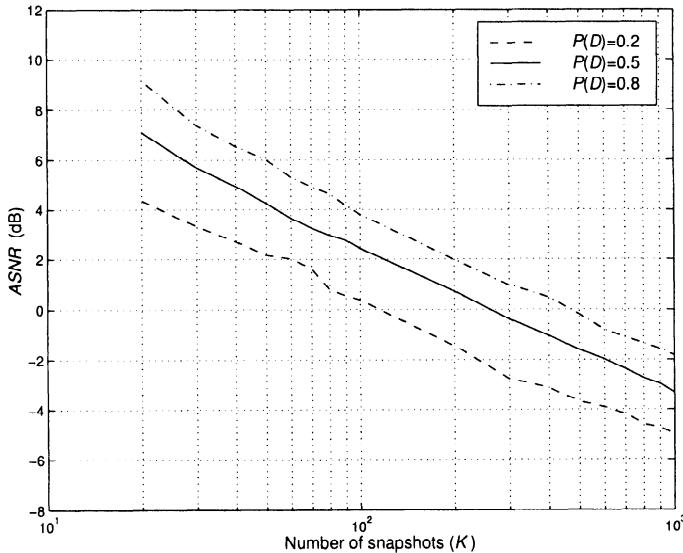


Figure 7.53 AIC algorithm; two equal-power signals,  $\Delta u = 0.5$  HPBW,  $P_D = 0.2$ , 0.5, and 0.8;  $ASNR$  detection threshold versus  $K$ .

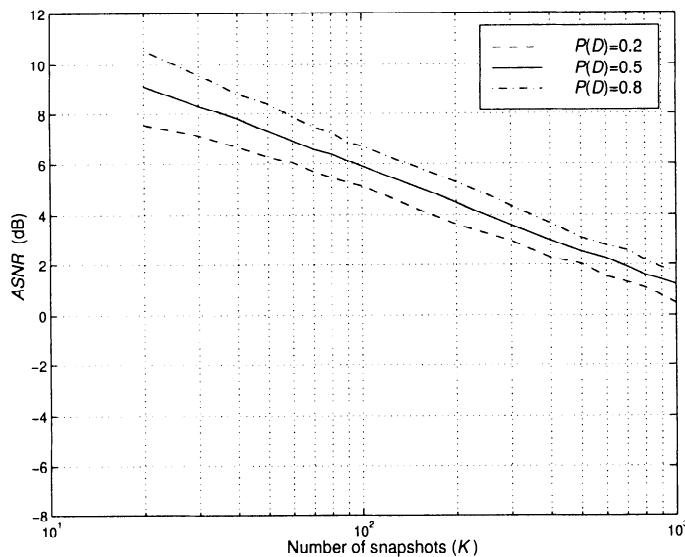


Figure 7.54 MDL algorithm; two equal-power signals;  $\Delta u = 0.5$  HPBW,  $P_D = 0.2, 0.5$ , and  $0.8$ , ASNR detection threshold versus  $K$ .

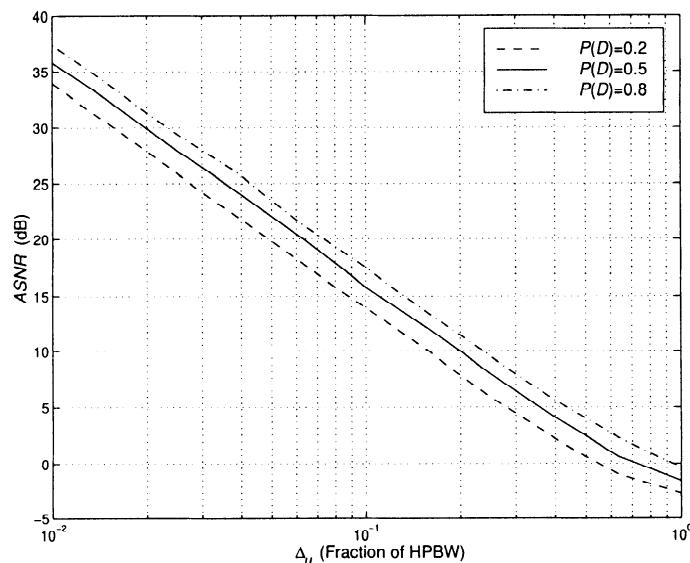


Figure 7.55 AIC-FO algorithm; two equal-power signals,  $K = 100$ ; detection threshold versus  $\Delta u/\text{HPBW}$ .

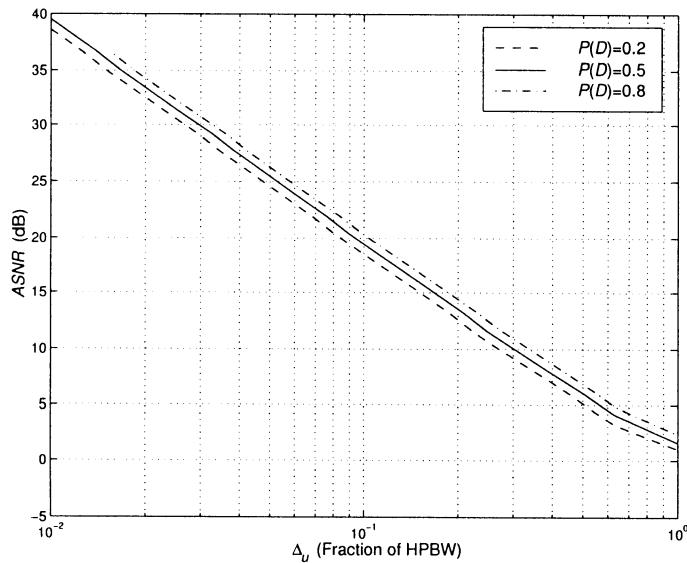


Figure 7.56 MDL-FO algorithm; two equal-power signals,  $K = 100$ ; detection threshold versus  $\Delta u/\text{HPBW}$ .

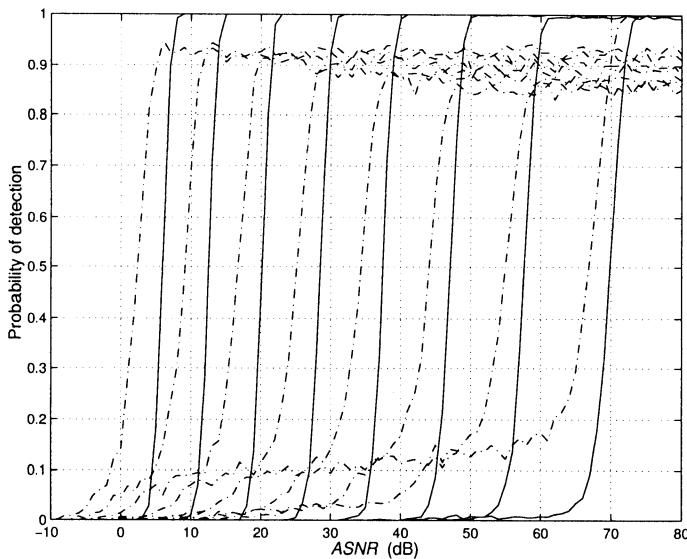


Figure 7.57 AIC-FO and MDL-FO algorithms; uniformly spaced, equal-power signals,  $\Delta u = 0.5$  HPBW,  $P_D$  versus ASNR for each signal,  $D = 2, \dots, 9$ .

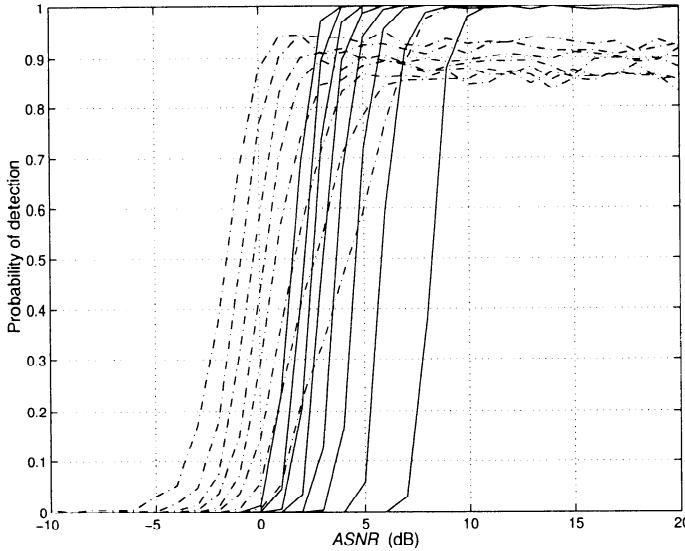


Figure 7.58 AIC-FO and MDL-FO algorithms; uniformly spaced, equal-power signals,  $\Delta u = 1.0$  HPBW,  $P_D$  versus  $ASNR$  for each signal,  $D = 2, \dots, 9$ .

7.9, we study the model in which we use  $\hat{d}$  to determine the dimension of an eigenspace beamformer. In Chapters 8 and 9, we use  $\hat{d}$  as an input to various parameter estimation algorithms.

In this subsection, we used tests that utilized the estimated eigenvalues. In the next subsection, we develop tests that utilize the estimated eigenvectors.

### 7.8.2 Eigenvector Detection Tests

The tests in Section 7.8.1 utilized the eigenvalues of  $\mathbf{S}_x$  in their detection tests. When the signal separation is small or  $|\rho|$  is close to one, some of the signal eigenvalues are small enough that the tests are not reliable. Xu et al. [XPK92] introduced the idea of eigenvector tests. Lee and Li [LL94] developed an efficient technique that is useful for closely spaced signals. It exploits Lee's earlier work on eigenvalues and eigenvectors of closely spaced signals [Lee92] that we discussed in Section 5.5.4. We develop the eigenvector detection technique (EDT) in this section. Our discussion follows [LL94].

We assume that we have done a preliminary beamscan using a conventional beamformer or an MPDR beamformer. Several locations in  $u$ -space

where one or more signals appear to be present have been detected.<sup>29</sup> We refer to each of these locations as a cluster and denote the beam locations where we detected the clusters at  $\psi_{C1}, \psi_{C2}, \dots, \psi_{CL}$ . We focus our attention of a single cluster and want to develop a test to detect the number of signals in the cluster.

Denote the reference direction as  $\psi_0$  (this corresponds to one of the  $\psi_{Ci}$ ). If there are  $d$  signals in the cluster, we denote their location by  $\psi_1 < \psi_2 < \dots < \psi_d$ . To emphasize their close spacing, we write

$$\psi_i = \psi_0 + \Delta\psi_i, \quad (7.512)$$

and

$$\Delta\psi_i = \Delta\psi \cdot q_i, \quad (7.513)$$

where the  $q_i$  are real distinct constants,  $q_1 < q_2 < \dots < q_d$ , with  $q_1 = -1/2$  and  $q_d = +1/2$ , and  $\Delta\psi$  is the scale factor that defines the width of the cluster in  $\psi$ -space.

We define  $\Phi_1, \dots, \Phi_d$  as the limiting value of the eigenvectors as  $\Delta\psi \rightarrow 0$ . We calculate the spatial derivatives of  $\mathbf{v}(\psi)$  evaluated at  $\psi_0$ ,

$$\mathbf{v}^k(\psi_0) \triangleq \frac{d\mathbf{v}^k(\psi)}{d\psi}|_{\psi=\psi_0}, \quad (7.514)$$

and define

$$\dot{\mathbf{V}} = \begin{bmatrix} \mathbf{v}(\psi_0) & \mathbf{v}^1(\psi_0) & \dots & \mathbf{v}^{N-1}(\psi_0) \end{bmatrix}. \quad (7.515)$$

Then [Lee92] shows that the eigenvectors,  $\Phi_1, \dots, \Phi_d$ , are the first  $d$  columns of the matrix obtained by the left-to-right Gram-Schmidt orthonormalization of  $\dot{\mathbf{V}}$ .

We next compute the ordered eigenvectors of the sample covariance matrix. We denote these eigenvectors as  $\hat{\psi}_1, \hat{\psi}_2, \dots, \hat{\psi}_N$ . We then test successive eigenvectors,  $\hat{\psi}_i$ , for closeness to  $\Phi_i$ ,

$$\delta_i \triangleq |\hat{\psi}_i^H \Phi_i|^2. \quad (7.516)$$

If

$$\delta_i \geq T, \quad (7.517)$$

where  $T$  ( $T < 1$ ) is a threshold, we say that the eigenvector corresponds to a signal eigenvector. Thus,  $\hat{d}$  corresponds to the largest value of  $i$  that passes the threshold.

---

<sup>29</sup>We discuss detection algorithms in Chapter 10. However, in this case, the detection algorithms in DEMT III [VT71], [VT01b] are applicable.

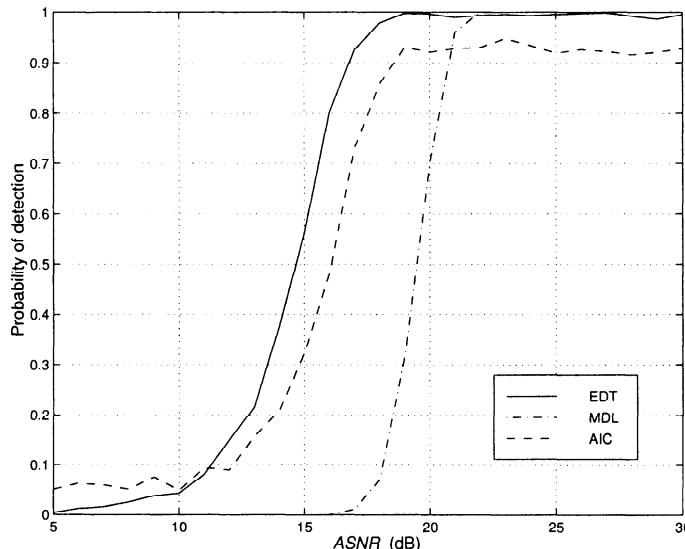


Figure 7.59 EDT, MDL, and AIC algorithms; two equal-power signals,  $\Delta u = 0.1$  HPBW,  $K = 100$ , 1,000 trials;  $P_D$  versus  $ASNR$ .

We consider a sequence of examples to illustrate the behavior of the EDT algorithm. The examples are similar to those in [LL94].

#### Example 7.8.5

Consider a standard 10-element linear array. Two equal-power uncorrelated signals impinge on the array from  $\pm\Delta\psi/2$ , where  $\Delta\psi$  is 0.1 HPBW (0.022 radians for this array). We use  $\psi_0 = 0$  and  $T = 0.5$ . We assume  $K = 100$ .

In Figure 7.59, we show  $P_D$  versus  $ASNR$  for the EDT algorithm. We also show the AIC and MDL results.

We see that the EDT is about 2 dB better than the AIC algorithm and its  $P_D$  approaches unity for  $ASNR \geq 20$  dB.

However, the EDT algorithm was given additional prior information. In the next example, we utilize that information to modify the AIC and MDL algorithms.

#### Example 7.8.6 (continuation)

The EDT algorithm assumed that a beamscan provided prior information regarding the centroid of the cluster. With this prior information we could utilize AIC or MDL in beampspace. We first define<sup>30</sup>

$$\mathbf{B}_{no}^H = \dot{\mathbf{V}}_3, \quad (7.518)$$

<sup>30</sup>We motivate this choice when we study parameter estimation in Chapter 8.

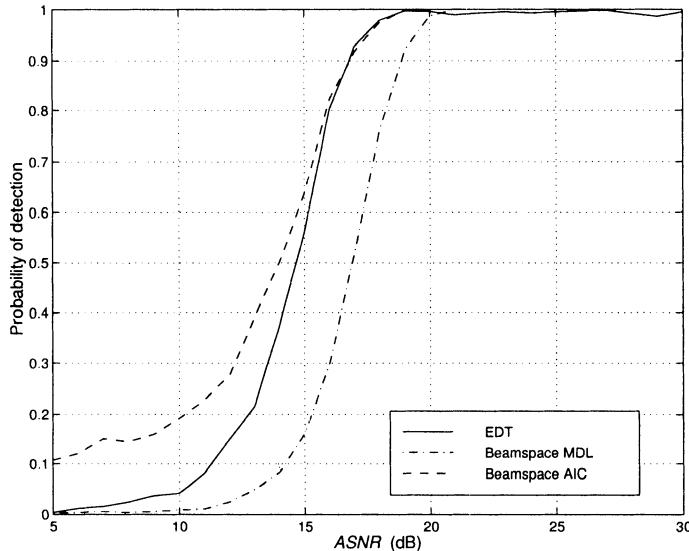


Figure 7.60 EDT, BS-AIC, and BS-MDL algorithms; two equal-power signals,  $\Delta u = 0.1\text{HPBW}$ ,  $K = 100$ , 1,000 trials,  $P_D$  versus  $ASNR$ .

where  $\dot{\mathbf{V}}_3$  is an  $N \times 3$  matrix containing the first three columns of  $\dot{\mathbf{V}}$ . Then, using (6.518),

$$\mathbf{B}_{bs}^H = [\dot{\mathbf{V}}_3^H \dot{\mathbf{V}}_3]^{-\frac{1}{2}} \dot{\mathbf{V}}_3^H. \quad (7.519)$$

In Figure 7.60, we plot  $P_D$  versus  $ASNR$ . We see that beamspace AIC performs better than EDT for  $P_D \leq 0.6$  and then the two algorithms have similar performance.

#### Example 7.8.7 (continuation, Example 7.8.5)

Consider the same model as in Example 7.8.5. We repeat the test for  $\Delta u = 0.5$  HPBW. The results are shown in Figure 7.61. For clarity, we show the results for the EDT, BS-AIC, and AIC algorithms. We see that, even with the larger spacing the EDT test performs better than the AIC tests.

The EDT algorithm provides an alternative approach to detecting the number of closely spread plane-wave signals that are present. It requires a preliminary processing step to detect the location of signal clusters. It appears that, if we utilize this preliminary processing, then beamspace AIC will have a similar performance.

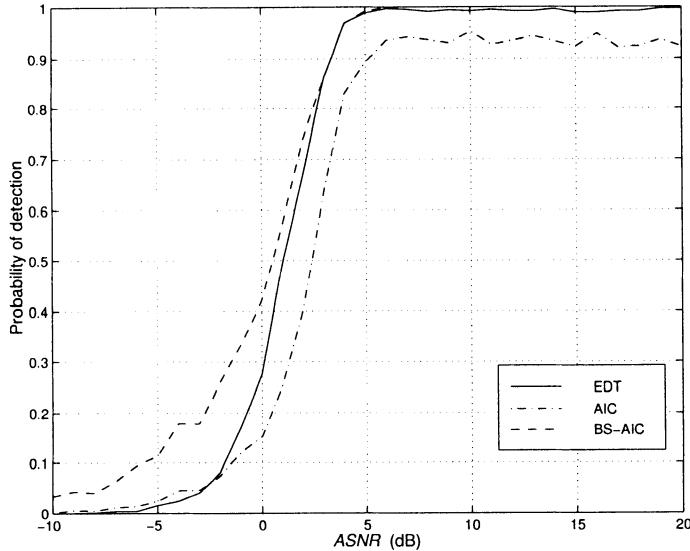


Figure 7.61 EDT, AIC, BS-AIC algorithms, two equal-power signals,  $\Delta u = 0.5$  HPBW,  $K = 100$ , 1,000 trials,  $P_D$  versus  $ASNR$ .

## 7.9 Eigenspace and DMR Beamformers

In Section 6.8, we discussed several beamformers that utilized an eigendecomposition to implement the beamformer. We assumed that the spatial spectral matrix,  $\mathbf{S}_x$ , is known. In this section, we consider the case in which  $\mathbf{S}_x$  is estimated from the data. We focus our attention on two beamformers, the principal-component (or eigenspace) beamformer of Section 6.8.1 and the dominant-mode rejection beamformer of Section 6.8.3. We restrict our discussion in the text to the model in which the desired signal and interference are linearly independent plane waves. For the eigenspace beamformers, (6.449) applies if we introduce the estimated quantities in place of the ensemble statistics. The MPDR eigenspace beamformer is

$$\mathbf{w}_{mpdr,es}^H = \gamma_{es} \mathbf{v}_m^H \hat{\mathbf{U}}_{S+I} \hat{\Lambda}_{S+I}^{-1} \hat{\mathbf{U}}_{S+I}^H, \quad (7.520)$$

where

$$\gamma_{es} = \left( \mathbf{v}_m^H \hat{\mathbf{U}}_{S+I} \hat{\Lambda}_{S+I}^{-1} \hat{\mathbf{U}}_{S+I}^H \mathbf{v}_m \right)^{-1}. \quad (7.521)$$

For the DMR beamformers, (6.491) applies if we use estimated quantities,

$$\mathbf{w}_{dm}^H = \frac{\alpha \sum_{i=1}^{\hat{D}_m} \frac{1}{\lambda_i} (\mathbf{v}_m^H \hat{\Phi}_i) \hat{\Phi}_i^H + \mathbf{v}_m^H \hat{\mathbf{P}}_{dm}^\perp}{\alpha \sum_{i=1}^{\hat{D}_m} \frac{1}{\lambda_i} |\mathbf{v}_m^H \hat{\Phi}_i|^2 + \mathbf{v}_m^H \hat{\mathbf{P}}_{dm}^\perp \mathbf{v}_m}. \quad (7.522)$$

In this section, we study the adaptive versions of these two beamformers. In Section 7.9.1, we study SMI implementations of adaptive eigenspace beamformers. In this case, we receive a block of data, compute the eigenvalues and eigenvectors, and implement the beamformers in (7.520) and (7.522). We compare their performance to the MPDR-SMI beamformer in Section 7.3 and the known-statistics results in Sections 6.8.1 and 6.8.3. In Section 7.9.1, we assume that the dimension of the  $S + I$  subspace is known.

In Section 7.9.2, we consider the SMI implementation when the dimension of the  $S + I$  subspace must be estimated from the data. In Section 7.9.3, we discuss the implementation in which we track the  $S + I$  subspace as new data arrive at the array. In Section 7.9.4, we summarize our results.

### 7.9.1 Performance of SMI Eigenspace Beamformers

There are several performance questions that are pertinent to the eigenspace beamformers:

- (i) SMI performance. How quickly do eigenspace beamformers converge to the asymptotic solution?
- (ii) How well do the eigenspace beamformers perform with signal mismatch ( $\mathbf{v}_a \neq \mathbf{v}_m$ )?
- (iii) How well do eigenspace beamformers perform in the presence of array perturbations?
- (iv) How well do eigenspace beamformers perform when the dimension of the signal subspace must be estimated?

We consider the first two questions in this section under the assumption that the dimension of the signal plus interference subspace is known. We consider the fourth question in Section 7.9.2.

Several analyses for the eigenspace beamformer have been developed using various series expansions and approximations. One approximation, which is derived by Chang and Yeh [CY92] (e.g., [FG94] and [YL96]), is

$$E[SINR_o] \simeq \frac{SINR_o K}{K + SINR_o \cdot (N_{eig} - 1)}$$

$$= \frac{SINR_o K}{K + SINR_o(N_I)}, \quad (7.523)$$

where  $SINR_o$  is the output  $SINR$  when  $\mathbf{S}_x$  is known,  $N_{eig}$  is the number of eigenvalues used, and  $N_I$  is the number of interferers.

In order to have the  $E[SINR_o]$  within 3 dB of  $SINR_o$ , we require

$$K = N_I \cdot SINR_o. \quad (7.524)$$

This result can be compared to (7.111) from Section 7.2.1,

$$K = (SINR_{mpdr})(N - 1) \simeq SINR_{mpdr}N. \quad (7.525)$$

Thus, the eigenspace algorithm has reduced the multiplier from  $N$ , the number of degrees of freedom to  $N_I$ , the number of interferers.

We can also use (7.110) with  $N$  replaced with  $N_{eig}$ ,

$$E[\eta] \simeq \frac{K - N_{eig} + 2}{K + 1} \left\{ \frac{1}{1 + SINR_{mpdr-es} \frac{N_{eig}-1}{K+2}} \right\}. \quad (7.526)$$

We consider a simple example to illustrate the behavior.

**Example 7.9.1: Eigenspace beamformer** (continuation, Example 7.3.2)

Consider a standard 10-element linear array. The signal arrives from  $u_s = 0$ . The  $SNR$  is 0 dB and 20 dB. Two equal-power uncorrelated interferers arrive at  $u_I = 0.29$  and 0.45, each with an  $INR = 20$  dB.

In Figure 7.62, we plot the average  $SINR_o$  versus  $K$  for the SMI-eigenspace beamformer with and without FB averaging. The horizontal line denotes the steady state value. We also show the MPDR results from Figure 7.5. We see that the eigenspace beamformer offers significant improvement. For low  $SNR$ , the analytic expression is accurate. As the  $SNR$  increases, the analytic expression underestimates the performance.

In this case, the improved performance is due to the reduced dimensionality of the subspace.

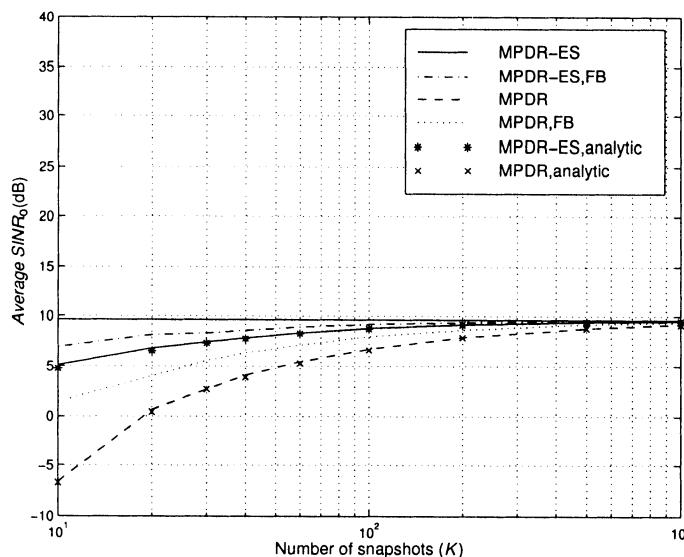
**Example 7.9.2: DMR beamformer** (continuation)

Consider the same model as in Example 7.9.1. In Figure 7.63, we plot the average  $SINR_o$  versus  $K$  for the SMI-DMR beamformer using  $D_m = 3$  with and without FB averaging. The  $SNR = 0$  dB in Figure 7.63(a) and 10 dB in Figure 7.63(b). The performance is several dB worse than the eigenspace beamformer. The estimates of the three dominant mode eigenvectors have some inaccuracy due to the finite data set. This mismatch causes some nulling of the signal.

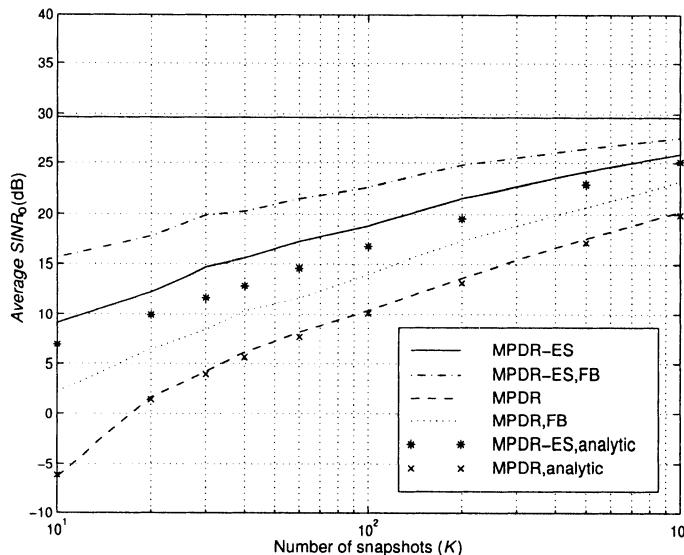
In the next two examples, we consider the case of signal DOA mismatch.

**Example 7.9.3: Eigenspace beamformer with mismatch** (continuation)

Consider the same nominal model as in Example 7.9.1. The beamformer assumes that the desired signal is arriving from  $u_s = 0$ . In practice, the signal arrives from  $u_a = -0.05$ .

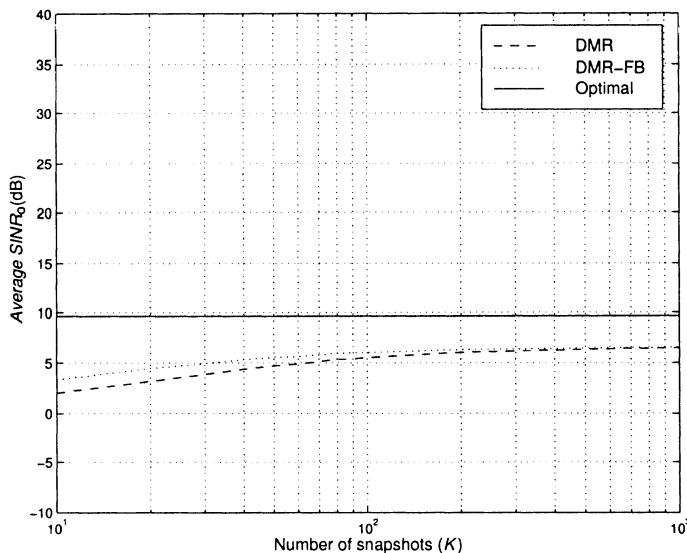


(a)

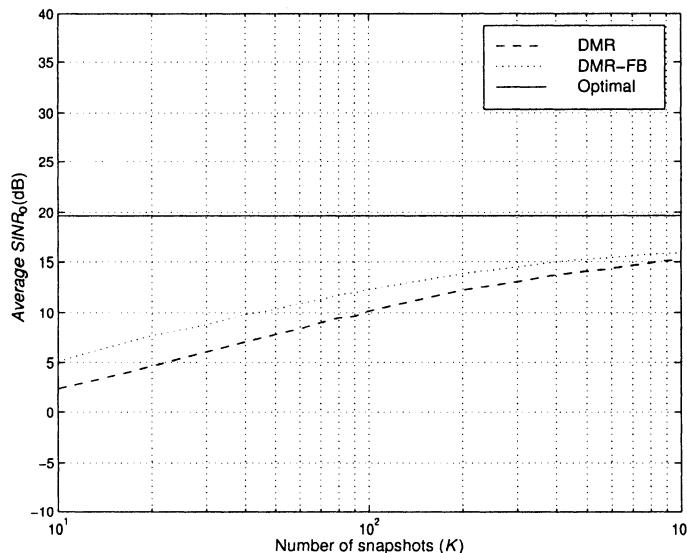


(b)

Figure 7.62 Eigenspace and MPDR beamformers using SMI algorithm:  $u_s = 0$ ,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB. 200 trials: (a)  $SNR = 0$  dB; (b)  $SNR = 20$  dB, average  $SINR_o$  versus  $K$ .



(a)



(b)

Figure 7.63 DMR beamformer with known subspace dimension using SMI algorithm:  $u_s = 0$ ,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB: (a)  $SNR = 0$  dB; (b)  $SNR = 10$  dB, average  $SINR_o$  versus  $K$ .

In Figure 7.64, we plot the performance of the eigenspace and MPDR beamformers. We see that there is dramatic improvement over the element space results. This improvement is because the algorithm is projecting the steering vector onto the eigenspace.

#### **Example 7.9.4: DMR beamformer with mismatch (continuation)**

Consider the same model as in Example 7.9.3. In Figure 7.65, we plot the same results as in Figure 7.64 for a DMR beamformer with  $\hat{D}_{dm} = 3$ . Because of the mismatch, a significant portion of the signal power is rejected.

In Figure 7.66, we plot the same results for a DMR beamformer with  $\hat{D}_{dm} = 2$ . The performance is much better because most of the signal power is not contained in the DM subspace when the *SNR* is much smaller than the *INR*. For  $SNR \geq INR$ , the performance is poor because a large portion of the signal is in the DM subspace and only a small portion of the interferer is in the DM subspace.

These results assume that the dimension of the eigenspace is known. In the next subsection, we consider the more realistic case in which the dimension of the eigenspace must be estimated.

### **7.9.2 Eigenspace and DMR Beamformers: Detection of Subspace Dimension**

In an actual application, the eigenspace or DMR beamformer must detect the dimension of the eigenspace as part of the implementation. We assume that the beamformer processes the input using one of the detection algorithms in Section 7.8 (e.g., AIC, MDL, AIC-FB, or MDL-FB). It assumes that the estimate,  $\hat{d}$ , is correct and implements the eigenspace or DMR beamformer in (7.520) or (7.522).

We illustrate the behavior by revisiting Example 7.9.1.

#### **Example 7.9.5**

Consider the same model as in Example 7.9.1. We implement the four tests from Section 7.8, AIC-FB, MDL-FB, AIC, and MDL. We then implement the MPDR-ES-FB or the MPDR-ES eigenspace beamformer. In Figures 7.67–7.69, we plot the average *SINR<sub>o</sub>* versus  $K$  for *SNR* = 0, 10, and 20 dB in part (a). In part (b) of each figure we plot  $P_D$ ,  $P_{FA}$ , and  $P_M$  versus  $K$ .

For the FB algorithms, the performance is essentially the same as the “known subspace dimension” result (from Example 7.9.1) for  $K \geq 10$ . For the FO algorithms, the performance is essentially the same as the “known subspace dimension” result for  $K \geq 20$ .

For the above values of *SNR* (0, 10, and 20 dB), the detection performance does not change significantly when the signal DOA is mismatched. Therefore, the results in Example 7.9.3 apply to the case of signal mismatch when the subspace dimension must be estimated.

In order for the detection algorithm to degrade the eigenspace beamformer, the *SNR* must be low enough to cause  $P_M$  to be significant. This

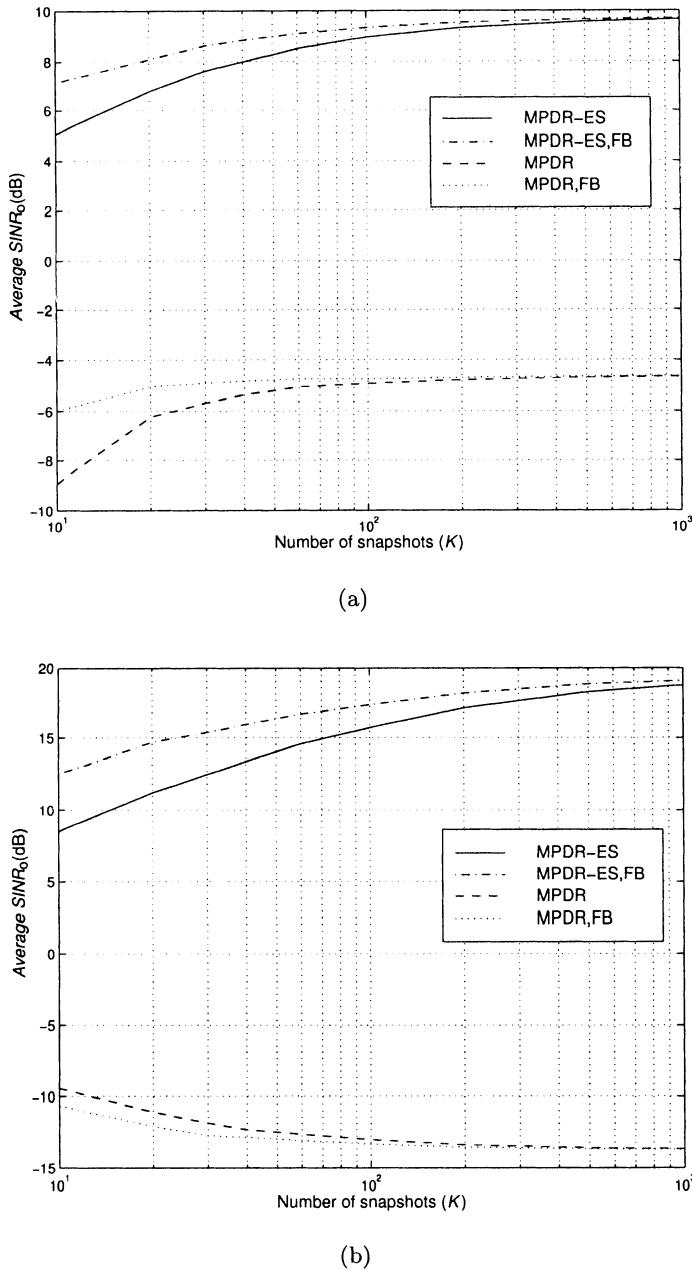


Figure 7.64 Eigenspace beamformer with signal mismatch:  $u_s = 0, u_a = -0.05, u_I = 0.29, 0.45, \text{INR} = 20 \text{ dB}, 200 \text{ trials}$ : (a)  $\text{SNR} = 0 \text{ dB}$ ; (b)  $\text{SNR} = 10 \text{ dB}$ ; average  $\text{SINR}_o$  versus  $K$ .

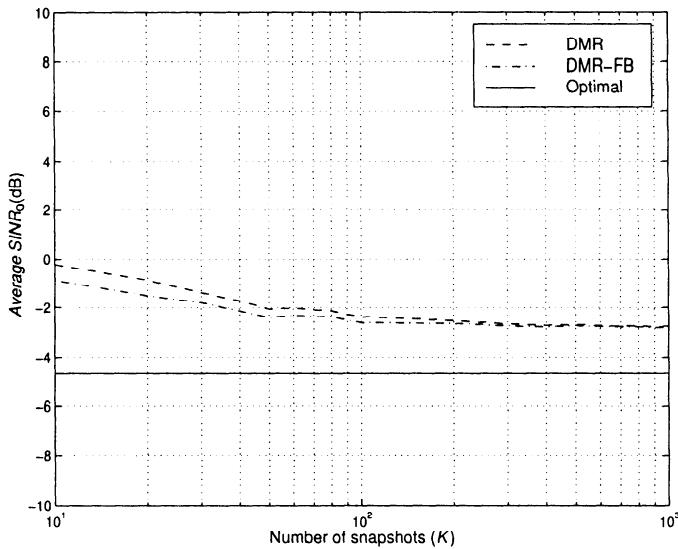


Figure 7.65 DMR beamformer with known subspace dimension and signal mismatch:  $\hat{D}_m = 3$ ,  $u_s = 0$ ,  $u_d = -0.05$ ,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB,  $SNR = 0$  dB; average  $SINR_o$  versus  $K$ .

result occurs when one of the eigencomponents that should be in the signal-plus-interference subspace is mistakenly assigned to the noise subspace. This result may be caused by a low  $SNR$ , but it can also be caused by the signal geometry. The impact of losing an eigencomponent depends on how closely it is correlated with the signal vector. We consider a case that stresses the algorithm in the next example.

#### Example 7.9.6 (continuation, Example 7.9.5)

Consider the same model as in Example 7.9.5. We assume that  $SNR = -10$  dB. We repeat the same tests as in Example 7.9.5. In Figure 7.70 we plot the average  $SINR_o$  versus  $K$  for the MPDR-ES-FB beamformer. In Figure 7.71, we plot  $P_D$ ,  $P_M$ , and  $P_{FA}$  versus  $K$ .

We see that there is significant performance degradation because, on many of the trials, we fail to detect the eigenvalue that is most correlated with the signal. The performance without FB averaging is much worse.

We next consider DMR beamformers. In Example 7.9.5, for  $SNR=0$ ,  $10$ , and  $20$  dB, the performance when the dimension is estimated is essentially the same as the known dimension case. Therefore, the case of most interest is the  $SNR = -10$  dB case in Example 7.9.6.

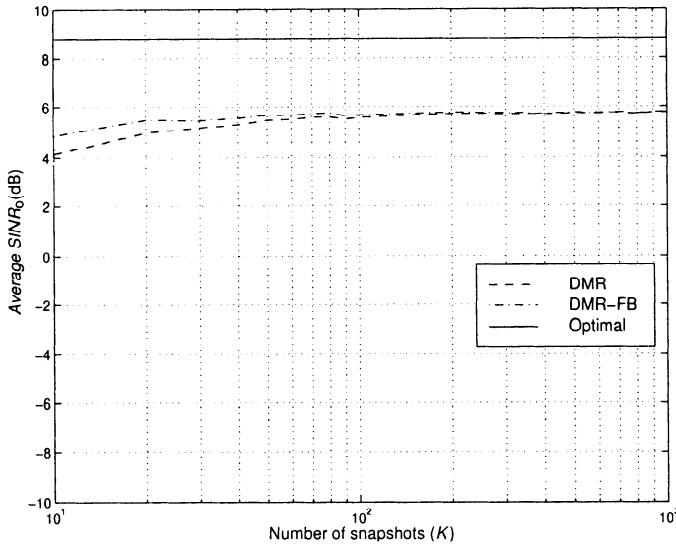


Figure 7.66 DMR beamformer with known subspace dimension and signal mismatch:  $\hat{D}_m = 2$ ,  $u_s = 0$ ,  $u_d = -0.05$ ,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB,  $SNR = 0$  dB; average output  $SINR_o$  versus  $K$ .

#### Example 7.9.7 (continuation)

Consider the same model as in Example 7.9.6. The detection algorithm results in Figure 7.71 also apply to this case. We implement the DMR beamformer with  $d_{dm} = \hat{d}$  where  $\hat{d}$  is the dimension estimate from one of the four detection algorithms. We use both FB and forward-only averaging. The results in Figure 7.71 show that, for small  $K$ , the detection algorithms are underestimating the dominant-mode subspace. Our results in Example 6.8.4 (see Figure 6.79) indicate that DMR should work well in this region. For  $K > 100$ , AIC-FB either correctly estimates  $d_{dm}$  or overestimates  $d_{dm}$ . For  $K > 200$ , MDL-FB correctly estimates  $d_{dm}$ . However, there is no signal mismatch, so the degradation due to having the signal in the DM subspace should be minimal.

In Figure 7.72, we plot the average  $SINR_o$  versus  $K$  for the four detection algorithms. We see that the MDL-DMR-FB algorithms are close to steady state by  $K = 20$ . The DMR algorithm using MDL-DMR-FO detection is close to steady state by  $K = 30$  because of its underestimation behavior: The DMR algorithm using AIC-FO converges much more slowly because of its overestimation behavior. For  $K < 100$ , the MPDR-DMR-FB algorithms perform better than the MPDR-FB algorithm.

For higher  $SNR$ , the detection results in Figures 7.67, 7.68, and 7.69 are applicable. We see that, for  $K > 20$  ( $2N$ ), the MDL-FB  $P_D$  is close to one. Therefore, the results in Example 7.9.2 should be close to the actual performance. Including the signal in the DM subspace causes performance degradation. When the  $SNR$  is less than the  $INR$ , we try an approach with

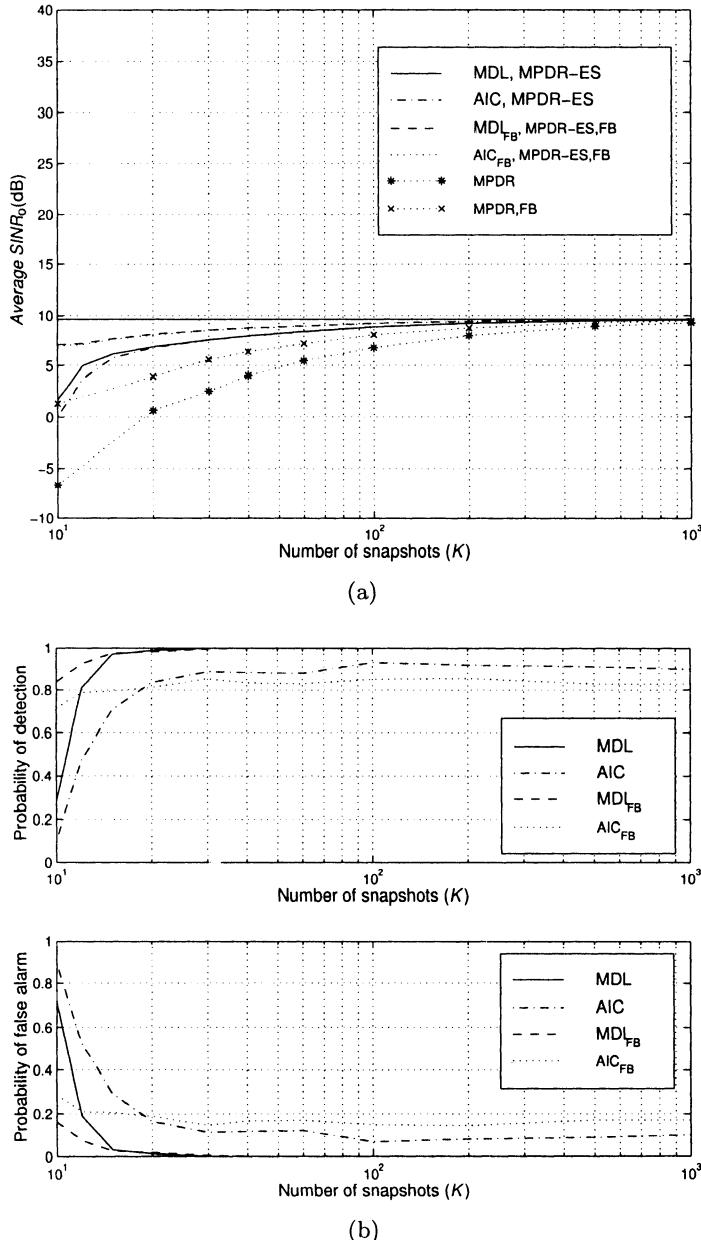


Figure 7.67 Eigenspace Beamformers using AIC and MDL detection algorithms;  $u_s = 0$ ,  $SNR = 0$  dB,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB, 200 trials: (a) average  $SINR_o$  versus  $K$ ; (b)  $P_D$  and  $P_{FA}$  versus  $K$ .

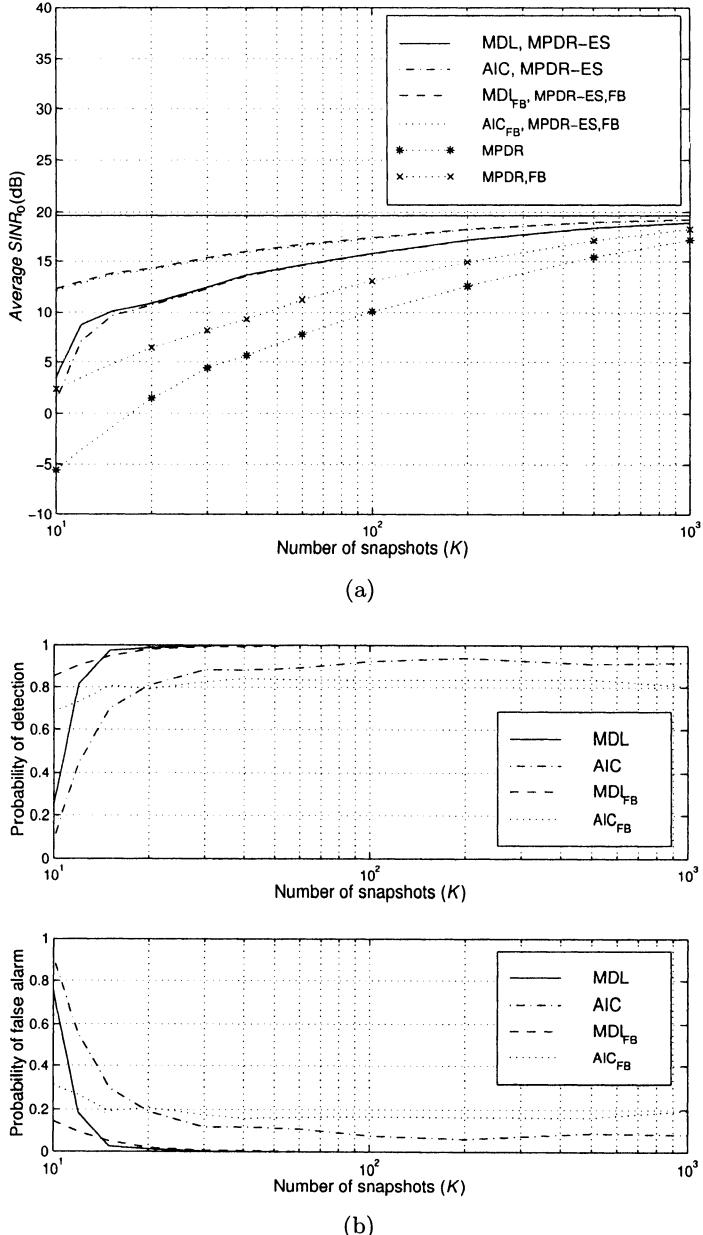


Figure 7.68 Eigenspace Beamformers using AIC and MDL detection algorithms:  $u_s = 0$ ,  $SNR = 10$  dB,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB, 200 trials: (a) average  $SINR_o$  versus  $K$ ; (b)  $P_D$  and  $P_{FA}$  versus  $K$ .

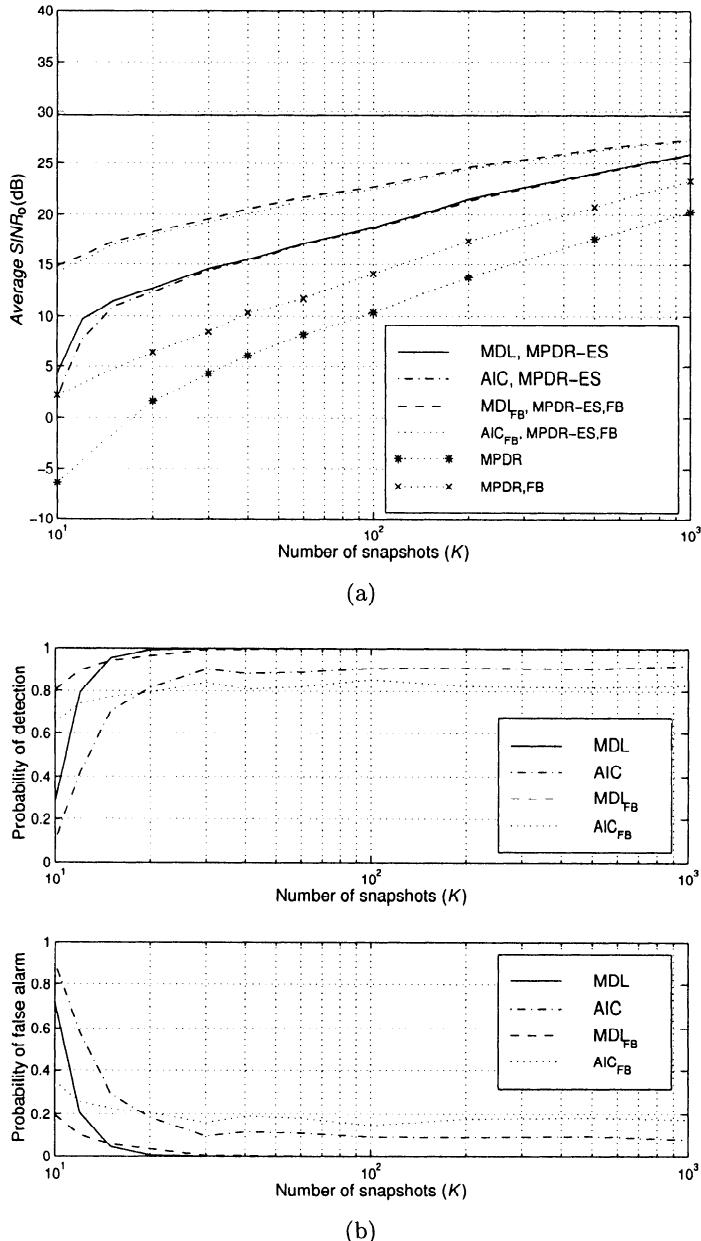


Figure 7.69 Eigenspace beamformers using AIC and MDL detection algorithms:  $u_s = 0$ ,  $SNR = 20$  dB,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB, 200 trials: (a) average  $SINR_o$  versus  $K$ ; (b)  $P_D$  and  $P_{FA}$  versus  $K$ .

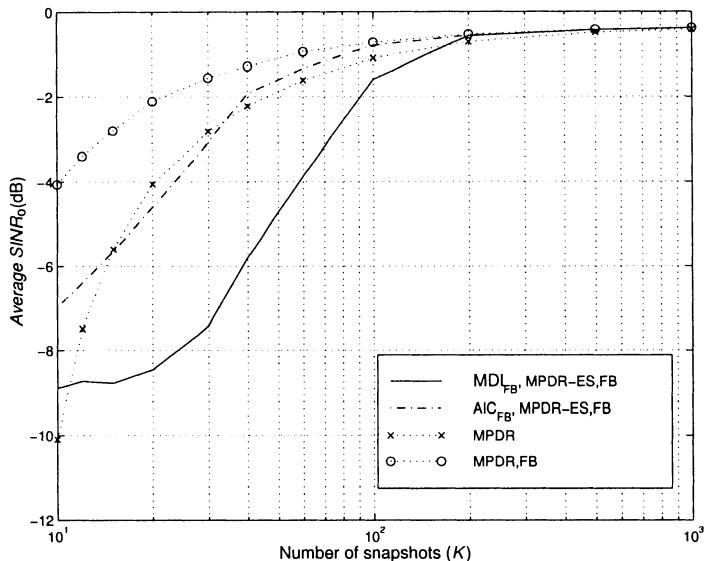


Figure 7.70 Eigenspace beamformers using AIC and MDL detection algorithms:  $u_s = 0$ ,  $SNR = -10$  dB,  $u_I = 0.29, 0.45$ ,  $INR = 20$  dB, 500 trials: average  $SINR_o$  versus  $K$ .

$d_{dm} = \hat{d} - 1$  to improve performance.

#### Example 7.9.8 (continuation)

Consider the same model as in Example 7.9.5 with  $SNR = 10$  dB. We implement the DMR algorithm with  $d_{dm} = \hat{d} - 1$  (and  $d_{dm} = \hat{d}$  for comparison). To simplify the figure, we only show the results for MDL-FB and AIC-FB. In Figure 7.73 we plot the average  $SINR_o$  versus  $K$  for an  $SNR = 10$  dB. We see that, for  $K < 500$ , deliberately underestimating the dimension of the DMR subspace improves the performance. For  $K < 100$ , the improvement in this example exceeds 3 dB.

In this section, we have studied the performance of eigenspace and DMR beamformers for the case in which we had to detect the dimension of the signal-plus-interference subspace. We used the AIC and MDL tests that were derived in Section 7.8 to detect the dimension.

For adaptive eigenspace beamformers, it is essential that the signal be included in the eigenspace. Therefore, for low  $SNR$  and small sample support we use the AIC-FB algorithm because it tends to overestimate, and the degradation in performance due to overestimation (false alarm) is less than the degradation due to underestimation (miss).

For higher  $SNR$ , AIC-FB still performs better, but the probability of

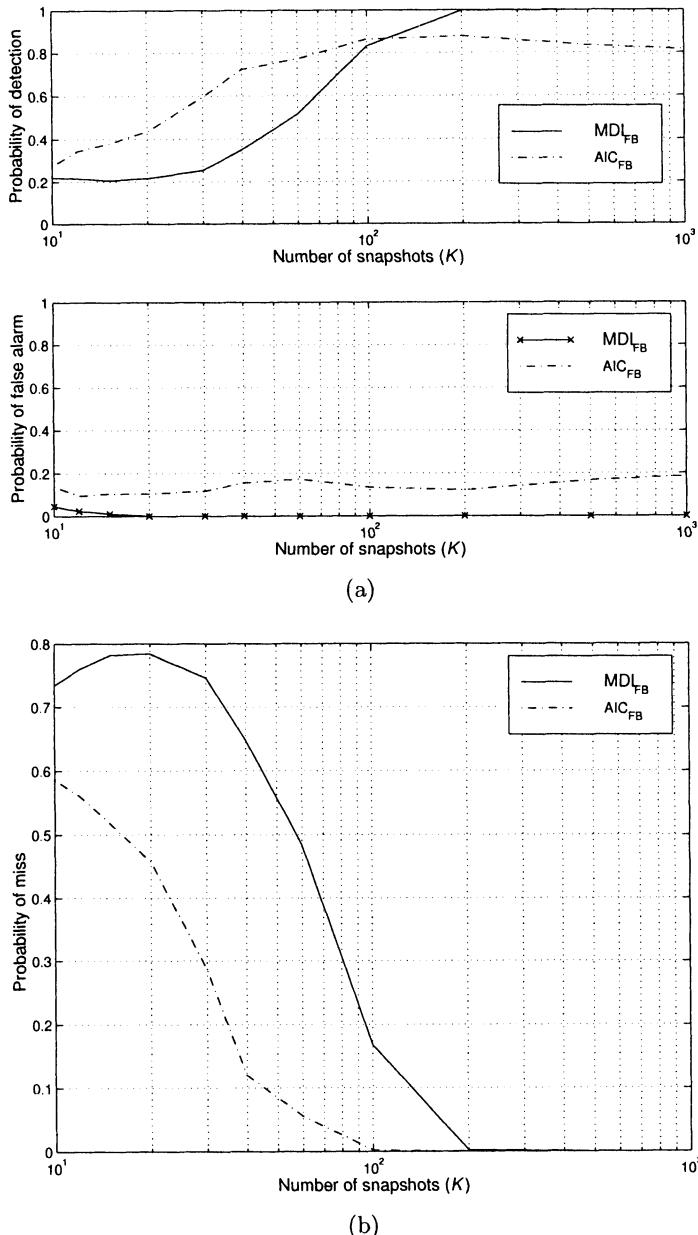


Figure 7.71 Performance of detection algorithms:  $u_s = 0$ ,  $\text{SNR} = -10$  dB,  $u_I = 0.29, 0.45$ ,  $\text{INR} = 20$  dB, 500 trials: (a)  $P_D$  and  $P_F$  versus  $K$ ; (b)  $P_M$  versus  $K$ .

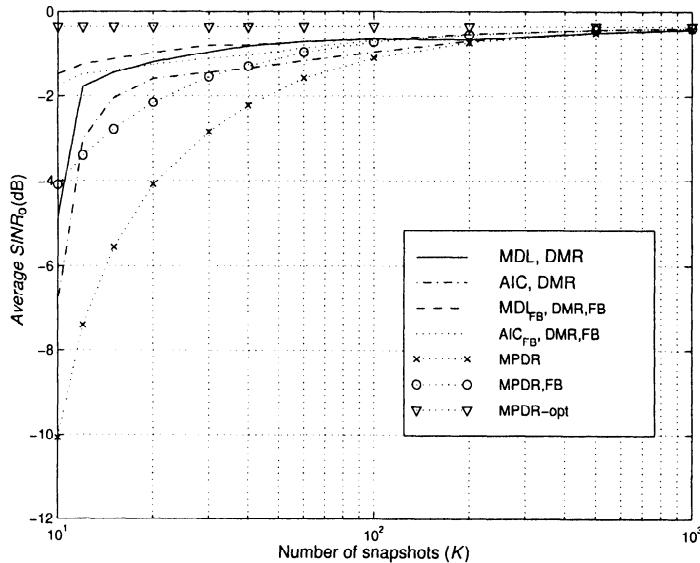


Figure 7.72 DMR beamformer with AIC and MDL detection algorithms,  $d_m = \hat{d}$ :  $u_s = 0$ ,  $SNR = -10$  dB,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB, 500 trials; average  $SINR_o$  versus  $K$ .

correct detection increases. As the  $SNR$  increases, the convergence is slower because of the signal presence, but the performance is always better than MPDR.

For adaptive DMR beamformers, it is better to have the signal excluded from the DM subspace whenever the DM subspace is inexact (e.g., due to the signal mismatch or finite data estimates). For low  $SNR$ , MDL-FB performs slightly better than AIC-FB because of its tendency to underestimate in the low sample support region. Cox and Pitre [CP97] and Cox et al. [CPL98] have developed a robust version of the DMR beamformer. These use an auxiliary test to decrease signal rejection.

These results indicate that the eigenspace algorithm is the best choice when the  $SNR$  is high enough that most of the signal power is included in the eigenspace. For low  $SNR$ , the DMR beamformer performs better. The detection algorithms in Section 7.8 that we use in this section used block processing. In the next section, we consider algorithms that track the subspace as new data arrives.

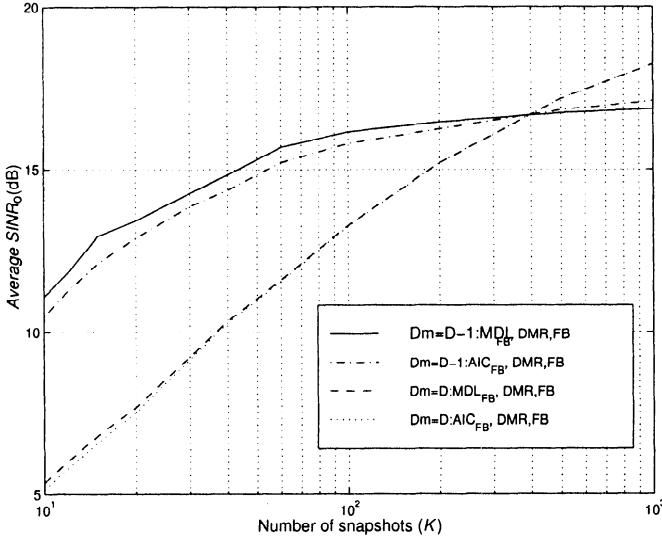


Figure 7.73 DMR beamformer with AIC and MDL detection algorithms,  $d_m = \hat{d} - 1$ :  $u_s = 0$ ,  $SNR = -10$  dB,  $u_I = 0.29$  and  $0.45$ ,  $INR = 20$  dB, 500 trials; average  $SINR_o$  versus  $K$ .

### 7.9.3 Subspace tracking

#### 7.9.3.1 Introduction

In Sections 7.9.1 and 7.9.2, the signal subspace is computed using a batch eigenvalue decomposition (ED) of the sample covariance matrix or using a SVD of the data matrix.<sup>31</sup> There are two disadvantages to this approach. The first is that the computational complexity is  $O(N^3)$ , where  $N$  is the number of sensors. In most cases of interest to us, we are only interested in the  $D$  signal eigenvalues and eigenvectors. Techniques have been developed to reduce the computational complexity. A discussion of these techniques is given in Tufts and Melissinos [TM86].

The second disadvantage is that, in adaptive beamforming (or parameter estimation) we would like to update the signal eigenvalues and eigenvectors of the sample covariance matrix (or the eigenvalues and singular vectors of the data matrix) each time a new data vector arrives. This updating is important because:

<sup>31</sup>This section is based on a research paper by one of my doctoral students, Zhi Tian [Tia99]

- (i) Each additional snapshot will improve the accuracy of our subspace estimate;
- (ii) In many applications of interest, the signal and/or interferers are moving so the subspace changes;
- (iii) New interfering signals may appear.

Any of these effects will cause the estimate of the subspace to change.

For these reasons, we would like to develop algorithms that track or update the eigendecomposition as new data vectors arrive. For adaptive eigenspace beamforming we need to track both the eigenvalues and eigenvectors. In Chapter 9, when we study parameter estimation, we will discuss algorithms (e.g., MUSIC) that utilize the subspace (defined by the eigenvectors), but do not need the eigenvalues. In this section, we restrict our attention to algorithms that track both the eigenvalues and eigenvectors.

We focus our attention on the case in which we have a signal and interference environment that is fixed over the observation interval. However, the algorithms can adapt to an additional interfering signal arriving or an existing interfering signal departing.

A second scenario of interest is the case in which the desired signal and interfering signals are moving and we want to track the subspace.

A number of adaptive algorithms for subspace tracking have been developed in the literature. A survey of work through 1990 is available in a paper by Comon and Golub [CG90]. More recent results are discussed in Dowling et al. [DAD94]. Most techniques can be put into three categories:

- (i) Variation of rank-one updating;

Golub [Gol73] developed an eigenvector (EV) updating scheme. Bunch et al. [BNS78] and Bunch and Nielsen [BN78] develop an algorithm that is guaranteed to converge. Karasalo [Kar86] provides an alternative implementation. DeGroat and Roberts [DR90] developed a numerically stable algorithm (ROSE) that provides good performance. MacInnes [Mac98] develops an update algorithm based on operator restriction (OPERA). Schreiber [Sch86] has developed implementations that use real operations. The FAST algorithm of Tufts et al. [TRC97] is another efficient example.

- (ii) Variations of QR factorizations;

This category includes various rank revealing decompositions, such as Stewart's URV algorithm [Ste92] and the TQR SVD of Dowling et al.

[DAD94]. Strobach [Str96] developed a low rank adaptive filter (LORAF) based on sequential orthogonal iteration (suggested by Owsley [Ows78]).

- (iii) Constrained or unconstrained optimization problem:

The eigenvector decomposition (EVD) or SVD can be formulated as an optimization problem and various techniques, such as gradient-based methods, Gauss-Newton iteration, and conjugate gradient techniques, can be applied to seek the largest or smallest eigenvalues and their corresponding eigenvectors adaptively. Yang and Kaveh [YK88] developed an algorithm using an LMS-type gradient estimator. Yang [Yan95a], [Yan95b] developed an RLS estimator called PAST.

From the computational point of view, we may distinguish among methods requiring  $O(N^2d)$ ,  $O(N^2)$ ,  $O(Nd^2)$ ,  $O(Nd)$  arithmetic operations every update, where  $N$  is the dimension of the input data vector and  $d$  is the dimension of the subspace that we are tracking. The  $O(N^2d)$  and  $O(N^2)$  techniques are of little practical interest in applications where  $N$  is much larger than  $d$ . The wide range of the computational complexity is mainly due to the fact that some algorithms update the complete eigenstructure, whereas others track only the signal or noise subspace.

We looked at a number of algorithms, but only studied three algorithms in detail:

- (i) Rank-one signal eigenstructure updating (ROSE), [DR90];
- (ii) Low rank adaptive filter (LORAF), [Str96];
- (iii) Projection approximation subspace tracking (PAST). [Yan95a], [Yan95b]

All of these algorithms are suitable for array processing<sup>32</sup> and are of complexity  $O(Nd^2)$  or  $O(Nd)$ .

In each case, we required the algorithm to estimate the dimension of the subspace and estimate the eigenvalues and eigenvectors. We found that ROSE and a slightly modified version of LORAF had the best performance. The reader is referred to the above references and [Tia99] for a detailed discussion. There are a number of other algorithms that may be useful in a particular application. These include:

- (i) Operator restriction algorithm (OPERA): [MacI98], [MV96];

---

<sup>32</sup>Some adaptive filter algorithms require a serial data input.

- (ii) Fast approximate subspace tracking (FAST): Tufts et al. [TRC97] and Real et al. [RTC99];
- (iii) Prony-Lanczos algorithm: Tufts and Melissinos [TM86];
- (iv) PC method: Champagne [Cha94];
- (v) Data-domain signal subspace updating algorithm (DDSSU): Youn and Un [YU94].

The reader is referred to the references for further discussions of these algorithms.<sup>33</sup>

A different approach uses a geometric entity referred to as a Grassmann manifold. Early work in this area is by Bucy [Buc91] and Smith [Smi93]. Further work is contained in Fuhrmann [Fuh98], Fuhrmann et al. [FSM96], and Srivastava [Sri00]. The reader is referred to these references for further discussion.

#### 7.9.4 Summary

In this section, we have discussed adaptive eigenspace and adaptive DMR beamformers. The first two sections utilized block processing. In Section 7.9.1, we assumed that the dimension of the signal-plus-interference subspace was known and demonstrated that the reduced dimension of the subspace resulted in improved adaptive performance compared to a full-rank ( $N$ ) beamformer. This improvement was significant if the ratio  $N/N_{eig}$  was large.

In Section 7.9.2, we considered the more realistic case in which the dimension of the subspace must be estimated. Guidelines for choosing an eigenspace or DMR beamformer were suggested.

The algorithms in Sections 7.9.1 and 7.9.2 are block algorithms. In Section 7.9.3, we identified several subspace tracking algorithms. In these algorithms, the eigendecomposition is updated each time a new data vector arrives. For a stationary environment, some of the algorithms provided the same performance as the block EVD or SVD. More importantly, they enabled us to track changes in the eigenvectors and eigenvalues.

Eigenspace and DMR algorithms provide effective performance in many scenarios and should be considered as a candidate design in many cases. In

---

<sup>33</sup>We recognize that our discussion of subspace tracking algorithms is too brief for the reader to understand the issues without extensive outside reading. In an earlier draft we derived the three algorithms, but the necessary discussion was long and not comprehensive enough.

the next section, we discuss adaptive beamformers operating in beamspace. This approach provides an alternative way to achieve the benefits of reduced-rank processing.

## 7.10 Beamspace Beamformers

We introduced beamspace beamformers in Section 3.10 in the context of classical array design. In Section 6.9, we developed optimum beamspace processors. In this section, we develop the beamspace versions of the various adaptive beamformers that we have developed in the earlier sections of the chapter. We use the model in Section 6.9.

The central result of this section is that, because of the reduced degrees of freedom in beamspace ( $N_{bs}$  versus  $N$ ), the adaptation will be faster. The performance will have a  $K/N_{bs}$  dependence in beamspace versus a  $K/N$  dependence in element space.

The basic relation of interest is

$$\mathbf{X}_{bs} = \mathbf{B}_{bs}^H \mathbf{X}, \quad (7.527)$$

where

$$\mathbf{B}_{bs}^H \mathbf{B}_{bs} = \mathbf{I}. \quad (7.528)$$

The beamspace array manifold vector is

$$\mathbf{v}_{bs}(\psi) = \mathbf{B}_{bs}^H \mathbf{v}_s(\psi). \quad (7.529)$$

The beamspace spatial spectral matrix is

$$\mathbf{S}_{\mathbf{x},bs} = \mathbf{B}_{bs}^H \mathbf{S}_{\mathbf{x}} \mathbf{B}_{bs}. \quad (7.530)$$

If the array manifold vector is conjugate symmetric and the columns of  $\mathbf{B}_{bs}$  are conjugate symmetric then, if we take the real part of the estimated beamspace matrix, it is equivalent to FB averaging of the data.

We define an  $N_{bs} \times K$  beamspace data matrix  $\tilde{\mathbf{X}}_{bs}$ , which has the same structure as (7.4). Then

$$\hat{\mathbf{S}}_{\mathbf{x},bs,fb} = \text{Re}[\tilde{\mathbf{X}}_{bs} \tilde{\mathbf{X}}_{bs}^H] = \mathbf{B}_{bs}^H \hat{\mathbf{S}}_{\mathbf{x},fb} \mathbf{B}_{bs}. \quad (7.531)$$

In addition,  $\mathbf{v}_{bs}$  is real in the conjugate symmetric case.

The requirement for conjugate symmetry of the rows of  $\mathbf{B}_{bs}^H$  is satisfied by many beamspace matrices used in practice. Therefore, we can use real arithmetic to implement the adaptive beamspace beamformer, which offers

significant computation savings. Note that  $\mathbf{X}_{bs}$  is complex so that we have to reformulate our algorithms in a manner similar to the algorithm in Section 7.4.6 in order to actually implement the processors using real arithmetic.

In Section 7.10.1, we consider SMI algorithms. In Section 7.10.2, we consider RLS algorithms. In Section 7.10.3, we consider LMS algorithms. In Section 7.10.4, we summarize our results.

### 7.10.1 Beamspace SMI

The SMI algorithms apply directly in beamspace with suitable notational changes. The examples parallel those in Section 7.3. In some cases, we include the element-space result for comparison. We first consider an MVDR beamformer with no diagonal loading.

The first example is beamspace SMI and is analogous to Example 7.3.1.

#### Example 7.10.1: Beamspace MVDR

Consider a standard 32-element linear array. The beamspace matrix is  $7 \times 32$  and uses conventional beams spaced at  $2/N$  intervals in  $u$ -space. The array is aimed at broadside. There is a single interferer with an  $INR = 10$  dB at  $u_I = 3/32$ .

We use an MVDR beamformer. In Figure 7.74, we plot the output  $SINR_o$  versus  $K$ , the number of snapshots. We also plot the analytic result for  $E[\rho]$  from (7.95) with  $N$  replaced by  $N_{bs}$ . In Figure 7.74(a) we plot versus  $K/N_{bs}$ .

In Figure 7.74(b), we plot  $SINR_o$  versus  $K/N$  and show the element-space results for comparison. As expected, the beamspace SMI converges more quickly to the steady state value.

In the next example, we consider the same model and study the MPDR beamformer.

#### Example 7.10.2: Beamspace MPDR (continuation)

Consider the same model as in Example 7.10.1. In Figure 7.75(a), we plot the beamspace and element space results for an  $SNR = 0$  dB. In Figure 7.75(b), we plot the same results for an  $SNR = 10$  dB. In Figure 7.75(c), we plot the same results for an  $SNR = 20$  dB.

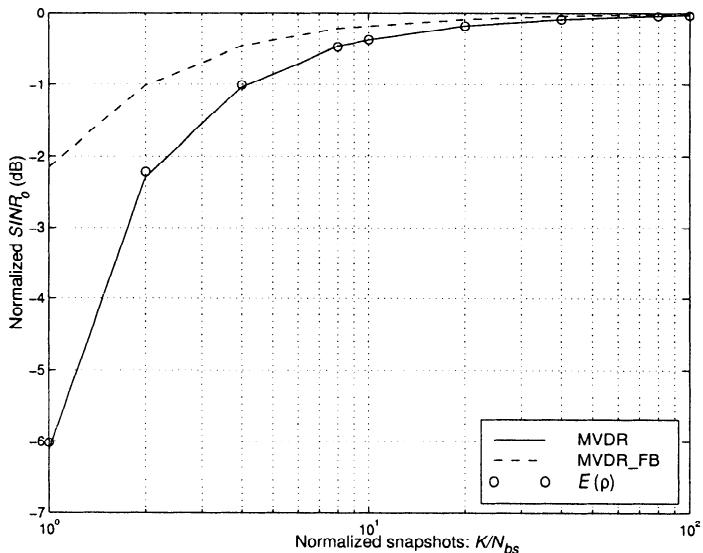
We see that there is a significant improvement in the performance. For the same level of  $SINR_o$  loss the ratio in the number of required snapshots is  $N_{bs}/N$ .

We next consider the use of fixed diagonal loading.

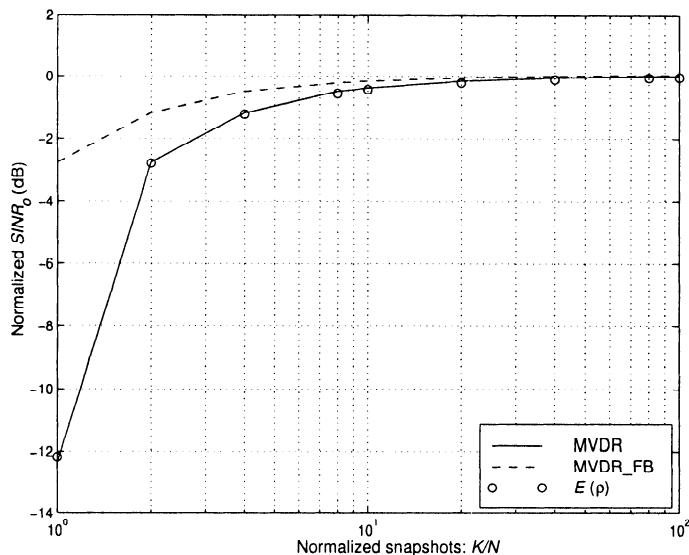
#### Example 7.10.3 (continuation)

Consider the same model as in Example 7.10.1 except the  $INR$  is increased to 20 dB. We consider two cases:

- (1)  $SNR = 0$  dB,  $INR = 15$  dB
- (2)  $SNR = 10$  dB,  $INR = 15$  dB



(a)



(b)

Figure 7.74 Beamspace MVDR with and without FB averaging:  $N = 32$ ,  $N_{bs} = 7$ ,  $u_s = 0$ ,  $u_I = 3/N$ ,  $INR = 10$  dB, 200 trials; normalized  $SNR$ : (a) versus  $K/N_{bs}$ ; (b) versus  $K/N$ .

In Figure 7.76(a), we plot the normalized  $SINR_o$  versus  $K/N_{bs}$  for case 1. In Figure 7.76(b), we plot representative beam patterns for  $K = 25$ . Figure 7.77 plots the same results for case 2. For both  $SNR$  levels, the loading provides a useful improvement in performance.

We next consider the signal mismatch problem. In Example 7.10.4, we use diagonal loading to combat signal DOA mismatch. In Example 7.10.5, we also use additional constraints.

**Example 7.10.4** (continuation)

Consider the same model as in Example 7.10.3. We use the beamspace MPDR with fixed diagonal loading,  $LNR = 20$  dB. There is a single interferer at  $u_I = 3/32$  with an  $INR = 20$  dB. The array is aimed at broadside but the signal arrives from  $u_a = -0.01$  with an  $SNR = 10$  dB.

In Figure 7.78, we plot the output  $SINR_o$  versus  $K/N_{bs}$ .

The next example of an SMI beamspace processor is a GSC implementation. The block diagram of the BS-GSC is shown in Figure 6.87. The equations specifying the BS-GSC are given in Section 6.9 ((6.524)–(6.528)). We consider an example using derivative constraints.

**Example 7.10.5: LCMP-BS-GSC-SMI beamformer** (continuation)

Consider the same model as in Example 7.10.4. Once again the array is aimed at broadside. The signal arrives from  $u_a = -0.01$  with an  $SNR = 10$  dB. There is a single interferer at  $u_I = 3/32$  with an  $INR = 20$  dB. We formulate the second-order derivative constraints in element space,

$$\mathbf{C} = [ \mathbf{1} \quad \dot{\mathbf{v}}(0) \quad \ddot{\mathbf{v}}(0) ], \quad (7.532)$$

and

$$\mathbf{g}^H = [ \mathbf{1} \quad \mathbf{0} \quad \ddot{\mathbf{B}}_c(0) ]. \quad (7.533)$$

Then, from (6.525),

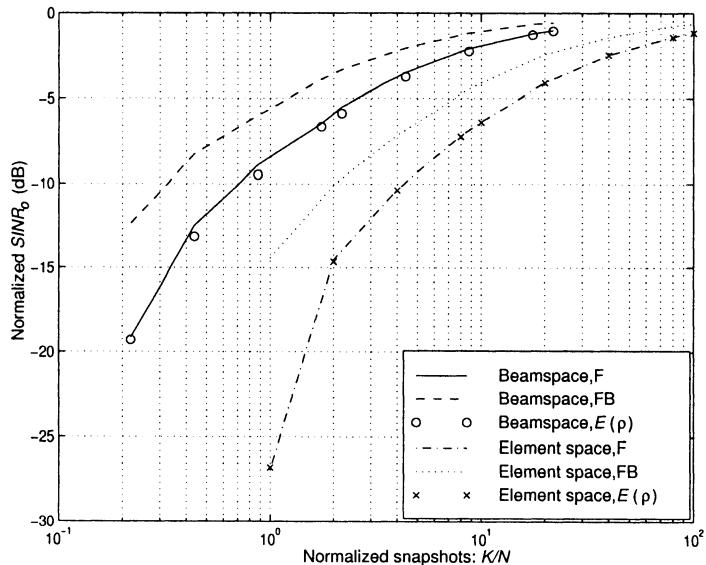
$$\mathbf{C}_{bs} = \mathbf{B}_{bs}^H \mathbf{C}, \quad (7.534)$$

and, from (6.528),

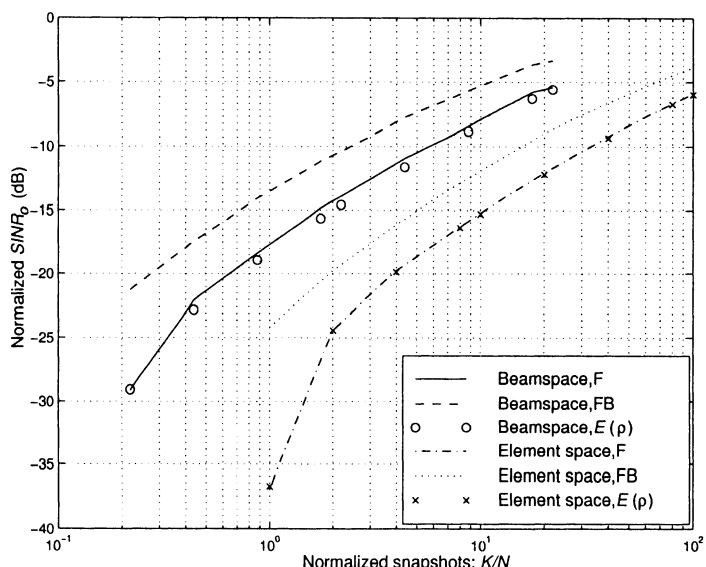
$$\mathbf{w}_{lcmp,bs}^H = \mathbf{g}^H [\mathbf{C}_{bs}^H \mathbf{S}_{\mathbf{x},bs}^{-1} \mathbf{C}_{bs}^H]^{-1} \mathbf{C}_{bs}^H \mathbf{S}_{\mathbf{x},bs}^{-1}. \quad (7.535)$$

We implement (7.535) using SMI with  $\mathbf{S}_{\mathbf{x},bs}$  replaced with  $\widehat{\mathbf{S}}_{\mathbf{x},bs,fb}$ .

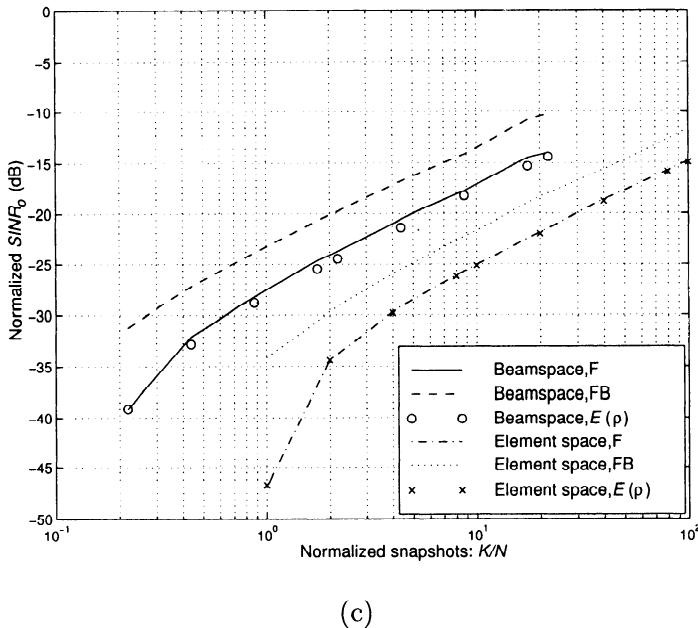
We add fixed diagonal loading of 17 dB. In Figure 7.79, we plot the average  $SINR_o$  versus  $K/N_{bs}$ . We can compare these results to those in Figure 7.78. With the BS-MPDR beamformer, we needed an  $LNR = 20$  dB to prevent signal nulling. With the LCMP-BS beamformer the derivative constraints helped prevent signal nulling, so an  $LNR = 17$  dB is adequate. The resulting steady state  $SINR_o$  was 3 dB higher and the convergence rate was similar to the MPDR case. We see that the additional constraints improve the mismatch performance.



(a)



(b)



(c)

Figure 7.75 Beamspace MPDR with and without FB averaging:  $N = 32$ ,  $N_{bs} = 7$ ,  $u_s = 0$ ,  $u_I = 3/N$ ,  $INR = 10$  dB, 200 trials; normalized  $SINR_o$  versus  $K/N$ : (a)  $SNR = 0$  dB; (b)  $SNR = 10$  dB; (c)  $SNR = 20$  dB.

In this section, we have studied SMI implementations of various beamspace beamformers. The key results are:

- (i) The convergence to steady state depends on  $N_{bs}$ , the dimension of the beamspace instead of  $N$ , the number of array elements.
- (ii) The computational complexity depends on  $N_{bs}$ , instead of  $N$ .

Most of our examples have used either  $N = 10$  or  $N = 32$  because many of the results scale. In many applications,  $N$  is much larger so that  $N_{bs}/N$  is very small. Thus, the advantages in (i) and (ii) are more dramatic.

In practice, we steer the array across the space of interest in discrete steps and perform beamspace processing at each step.

### 7.10.2 Beamspace RLS

In this section, we discuss beamspace RLS. All of the algorithms in Sections 7.4 and 7.5 can be modified for beamspace implementation in a straightfor-

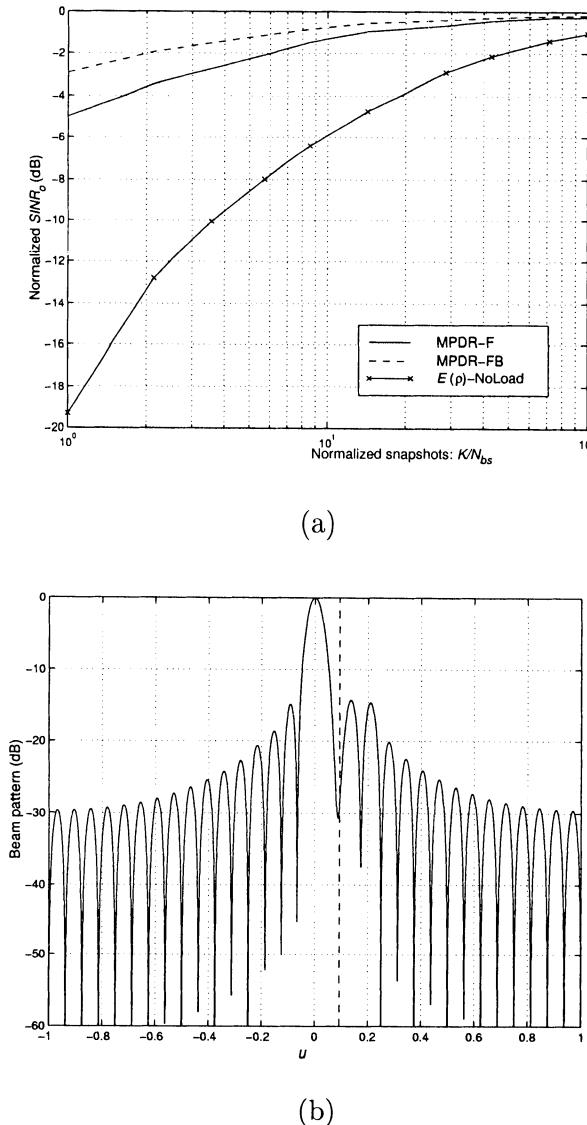


Figure 7.76 Beamspace MPDR with fixed loading:  $N = 32, N_{bs} = 7, u_s = 0, SNR = 0$  dB,  $u_I = 3/32, INR = 20$  dB,  $LNR = 15$  dB: (a) normalized  $SINR_o$  versus  $K/N_{bs}$ ; (b) typical beam patterns,  $K = 25$ :  $SNR = 0$  dB,  $INR = 20$  dB,  $LNR = 15$  dB.

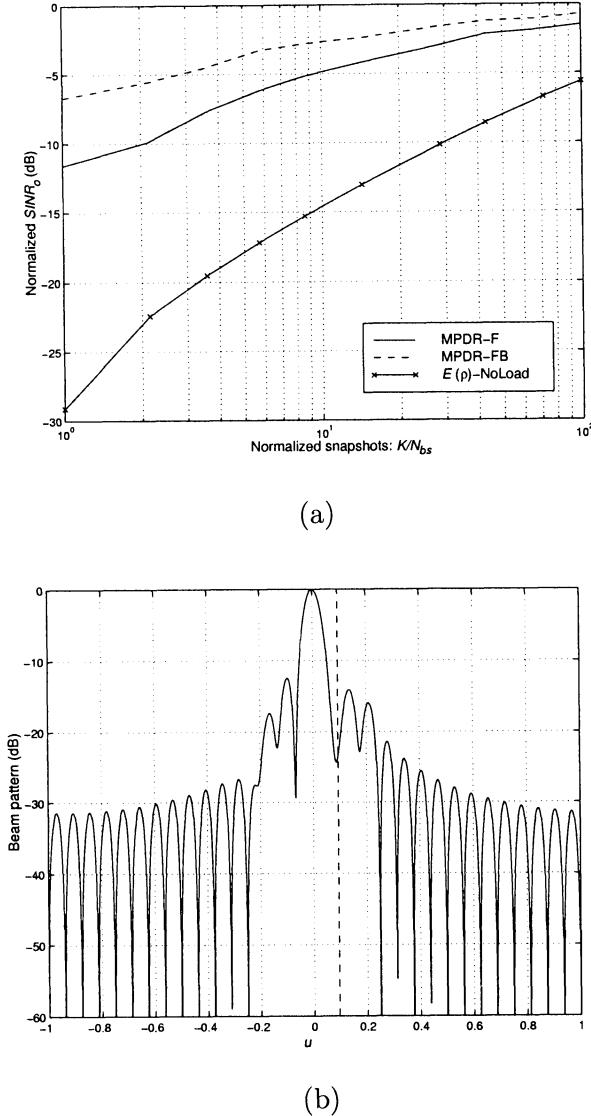


Figure 7.77 Beamspace MPDR with fixed loading:  $N = 32, N_{bs} = 7, u_s = 0, SNR = 10$  dB,  $u_I = 3/32, INR = 20$  dB,  $LNR = 15$  dB: (a) normalized  $SINR_o$  versus  $K/N_{bs}$ ; (b) typical beam patterns,  $K=25$ :  $SNR = 10$  dB,  $INR = 20$  dB,  $LNR = 15$  dB.

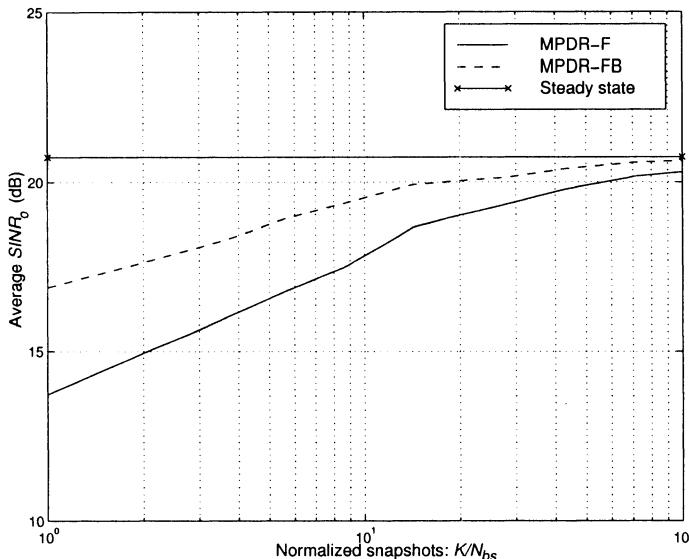


Figure 7.78 Beamspace MPDR with fixed loading and signal mismatch:  $N = 32$ ,  $N_{bs} = 7$ ,  $u_s = 0$ ,  $u_a = -0.01$ ,  $SNR = 10$  dB,  $u_I = 3/32$ ,  $INR = 20$  dB,  $LNR = 20$  dB; average  $SINR_o$  versus normalized snapshots,  $K/N_{bs}$ .

ward manner. In addition, if the array manifold vector and the rows of  $\mathbf{B}_{bs}^H$  are conjugate symmetric, then the algorithm in Section 7.4.6 allows us to use real computation.

To illustrate the behavior, consider an example using RLS with variable loading that was developed in Section 7.4.5. We use a BS-GSC implementation.

#### Example 7.10.6 (continuation, Example 7.10.4)

Consider the same model as in Example 7.10.4. We implement RLS-VL in beamspace with  $\mu=0.999$ . All of the other parameter values are specified in Example 7.10.4. For  $K/N_{bs} > 5$ , the RLS curves and the SMI are the same.

In practice, we can use the QRD to improve numerical stability. Other RLS examples are developed in the problems.

### 7.10.3 Beamspace LMS

In this section, we discuss beamspace LMS. All of the LMS algorithms in Section 7.7 can be modified for beamspace implementation in a straightforward manner. To illustrate the behavior we consider an example using

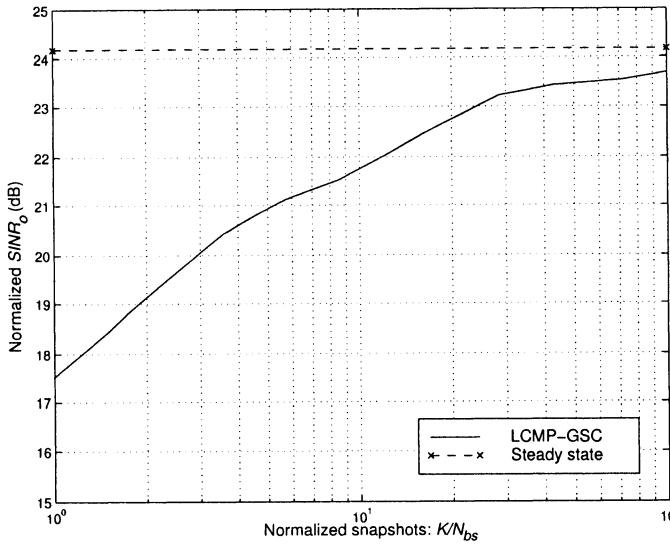


Figure 7.79 Beamspace LCMP beamformer with derivative constraints and fixed loading:  $N = 32, N_{bs} = 7, u_s = 0, u_a = -0.01, SNR = 10$  dB,  $u_I = 3/32, INR = 20$  dB,  $LNR = 17$  dB; average  $SINR_o$  versus normalized snapshots,  $K/N_{bs}$ .

LMS-SP in beamspace. The algorithm is described in Section 7.7.4.

**Example 7.10.7** (continuation, Example 7.10.4)

Consider the same model as in Example 7.10.4.  $N = 32$  and  $N_{bs} = 7$ . We implement LMS-SP in beamspace. We use  $T_o = 1.2/32$  and

$$\alpha(K) = \frac{\gamma}{\sigma_x^2(K)} \quad (7.536)$$

where

$$\sigma_x^2(K) = \beta\sigma_x^2(K-1) + (1-\beta)\mathbf{X}_{bs}^H(K)\mathbf{X}_{bs}(K) \quad (7.537)$$

(see (7.474) and (7.475)). In Figure 7.80, we plot the average  $SINR_o$  versus  $K/N_{bs}$  for several values of  $\gamma$  and  $\beta$ .

We see that the convergence is much faster than in element space.

#### 7.10.4 Summary: Adaptive Beamspace Processing

In this section, we have studied beamspace processing. Several important results are:

- (i) The convergence behavior of the algorithms is a function of  $K/N_{bs}$  instead of  $K/N$ . In most applications, the difference will be significant.

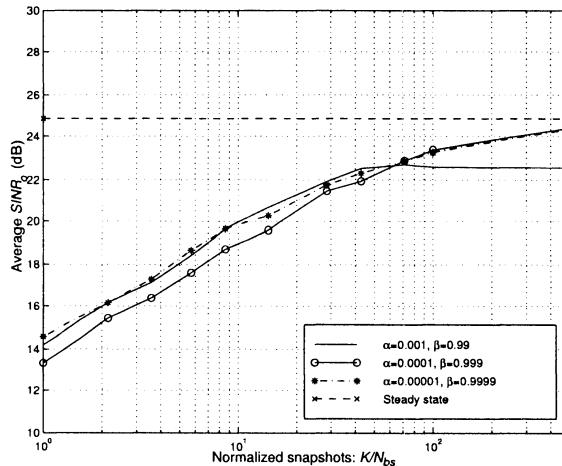


Figure 7.80 Beamspace MPDR beamformer using the LMS-SP algorithm:  $N = 32$ ,  $N_{bs} = 7$ ,  $u_s = 0$ ,  $u_a = -0.01$ ,  $SNR = 10$  dB,  $u_I = 3/32$ ,  $INR = 20$  dB,  $LNR = 17$  dB; average  $SINR_o$  versus normalized snapshots,  $K/N_{bs}$ .

- (ii) If the array is conjugate symmetric and the rows in the beamspace matrix are conjugate symmetric, then we can implement FB averaging by taking the real part of  $\hat{\mathbf{S}}_{\mathbf{x},bs}$ . This technique allows real processing to compute the necessary inverses. The result of (i) and (ii) can provide an order of magnitude improvement in the required number of snapshots to achieve a specified  $SINR_o$  loss from the steady state performance in many cases.
- (iii) All of the element space algorithms carry over to beamspace. However, the computational complexity is a function of  $N_{bs}$  rather than  $N$ .

As a result of these advantages, beamspace processors are widely used in practice.

## 7.11 Broadband Beamformers

In this section, we consider the adaptive behavior of the broadband beamformers that we developed in Section 6.13. We discussed a frequency-domain implementation and a time-domain implementation. The frequency-domain beamformer consisted of  $M$  uncoupled narrowband beamformers. Therefore, we can use the narrowband results in the earlier sections in this chapter to

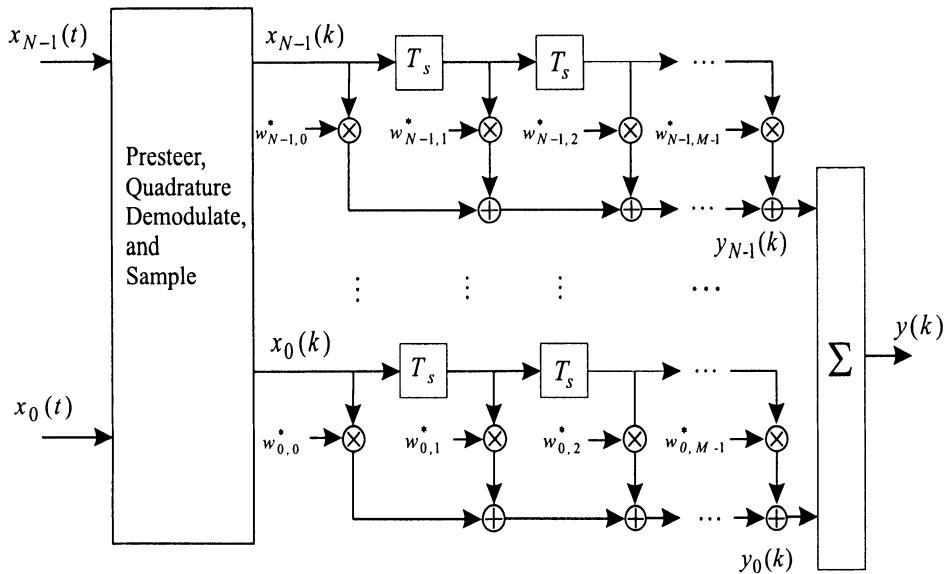


Figure 7.81 FIR beamformer: direct form.

analyze the broadband behavior. We consider several examples in the problems. In this section, we restrict our attention to broadband time-domain beamformers.

The starting point of the discussion is the time-domain model shown in Figure 7.81. This model is the direct form FIR model that we introduced in Figure 6.123.

The second time-domain model of interest is the GSC model in Figure 7.82 that we introduced in Figure 6.126. This GSC model assumes that the constraint matrix has the block structure in (6.797) or (6.810) and therefore,  $\mathbf{B}_{comp}^H$  has the sparse structure in (6.822).

In Section 7.11.1, we discuss the SMI implementation of the beamformers. In Section 7.11.2, we discuss the LMS implementation using the FIR filters in Figures 7.81 and 7.82. In Section 7.11.3, we discuss a lattice implementation of the FIR filter in order to improve the LMS convergence. In Section 7.11.4, we summarize our results.

### 7.11.1 SMI Implementation

In this section, we discuss the SMI implementation of the time-domain beamformer. We use the direct form model in Figure 7.81. The steady state

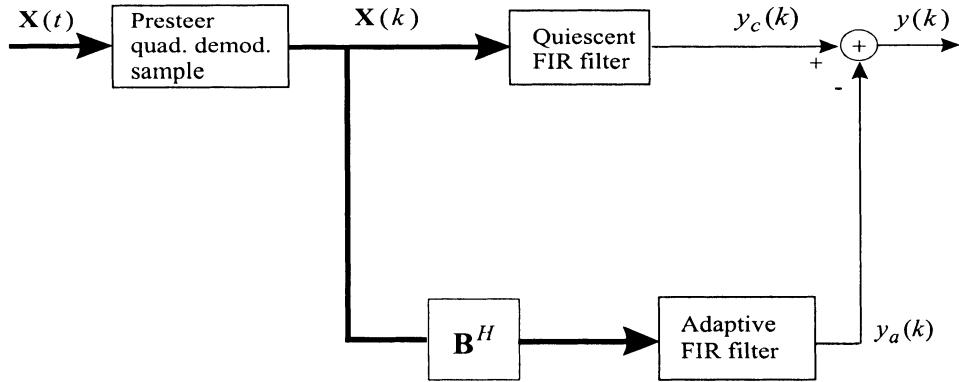


Figure 7.82 FIR beamformer: generalized sidelobe canceller.

equations are (from (6.806))

$$\mathbf{w}_{lcm}^H = \mathbf{g}^H \left[ \mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C} \right]^{-1} \mathbf{C}^H \mathbf{R}_x^{-1}, \quad (7.538)$$

where  $\mathbf{C}$  is an  $NM \times M_c M$  constraint matrix that has the block structure shown in (6.810).

For the MPDR case,

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}_M \otimes \mathbf{1}_N \quad (7.539)$$

and

$$\mathbf{g}^H = [1 \ 0 \ \cdots \ 0]. \quad (7.540)$$

In the SMI implementation, we estimate the  $NM \times NM$  matrix  $\mathbf{R}_x$  by constructing the sample covariance matrix,

$$\hat{\mathbf{R}}_x \triangleq \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k) \mathbf{x}^H(k), \quad (7.541)$$

where  $\mathbf{x}(k)$  is  $NM \times 1$  composite input vector defined in (6.761). As in the narrowband case, we add diagonal loading,

$$\tilde{\mathbf{R}}_x = \hat{\mathbf{R}}_x + \sigma_L^2 \mathbf{I}_{NM}. \quad (7.542)$$

We then compute the  $NM \times 1$  weight vector,

$$\mathbf{w}_{mpdr}^H = \mathbf{g}^H \left[ \mathbf{C}^H \left[ \hat{\mathbf{R}}_{\mathbf{x}} + \sigma_L^2 \mathbf{I}_{NM} \right]^{-1} \mathbf{C} \right]^{-1} \mathbf{C}^H \left[ \hat{\mathbf{R}}_{\mathbf{x}} + \sigma_L^2 \mathbf{I}_{NM} \right]^{-1}, \quad (7.543)$$

and

$$\hat{y}(K) = \mathbf{w}_{mpdr}^H \mathbf{x}(K). \quad (7.544)$$

We consider the same model as in Example 6.13.5.

**Example 7.11.1**<sup>34</sup> (continuation, Example 6.13.5)

Consider a uniform 10-element array with spacing  $d = \lambda_u/2$ . There is an  $M$ -tap FIR filter at the output of each sensor's quadrature demodulator. We implement a broadband MPDR beamformer using the SMI algorithm. The desired signal arrives from broadside ( $u_s = 0$ ) with a flat spectrum with  $B_f = 0.4$ . The interfering signal arrives from  $u_I = 0.29$  with an identical spectrum. The  $SNR = 0$  dB and the  $INR = 30$  dB.

In Figure 7.83, we plot the average  $SINR_o$  versus  $K$  for the  $M = 4$  case. Four curves are shown:

- (i)  $LNR = 0$  dB ( $SINR_{ss} = 9.24$  dB)
- (ii)  $LNR = 5$  dB ( $SINR_{ss} = 9.13$  dB)
- (iii)  $LNR = -50$  dB ( $SINR_{ss} = 9.26$  dB)
- (iv)  $\bar{\eta}(SINR_{ss})$ ; from (7.110), with  $LNR = -50$  dB and  $N = 40$ .

The algorithm with  $-50$ -dB loading has a slightly larger steady state  $SINR_o$ , but significantly poorer performance for  $K < 500$ . The behavior is accurately described by (7.110). The  $LNR = 0$ -dB and  $LNR = 5$ -dB cases have essentially the same steady-state but much better transient behavior. The  $LNR = 5$ -dB case is almost one dB higher than the  $LNR = 0$ -dB case at  $K = 100$ .  $LNR$ s greater than 5 dB lower the steady-state value without improving the transient behavior. Note that an  $LNR = 5$  dB is much lower than the  $LNR$  for the corresponding narrowband model.

In Figure 7.84, a representative beam pattern is plotted for  $K = 1,000$ . The nulls are in the correct location and the sidelobes are well-behaved.

In Figure 7.85, we plot the average  $SINR_o$  versus  $K$  for  $M = 2$ . The steady state value is 8.874 dB. At  $K = 1,000$ , the SMI algorithm is very close to the steady state value. In Figure 7.86, a representative beam pattern is plotted for  $K = 1,000$ . The nulls are in the correct location and the sidelobes are well-behaved.

The  $M = 2$  case has almost the same performance as the  $M = 4$  case, and the computational complexity is reduced. The reason for this behavior is that the signal and interferer have identical spectra, so the additional frequency resolution obtained by additional taps is not required.

As in the narrowband case, the SMI algorithm with appropriate loading provides good performance. The primary disadvantage is that the matrices that must be inverted are  $NM \times NM$ , which increases the computational complexity.

---

<sup>34</sup>The results in Examples 7.11.1 and 7.11.2 are due to J. Hiemstra and R. Jeffers (private communication).

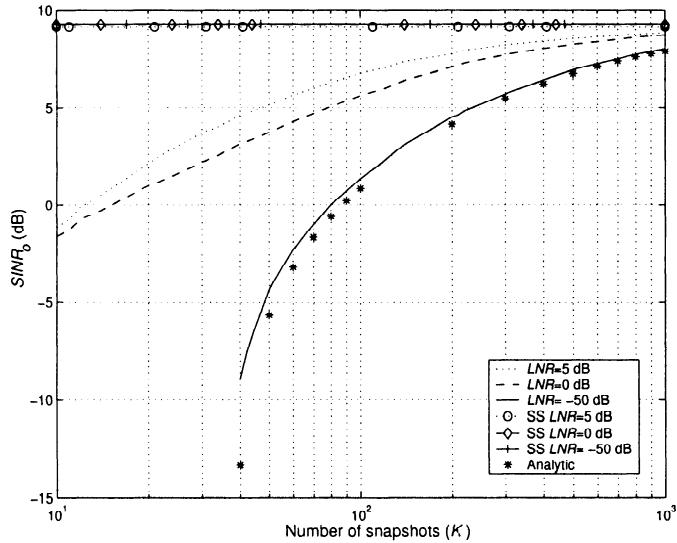


Figure 7.83 Broadband MPDR time-domain beamformer with SMI algorithm:  $M = 4, u_s = 0, SNR = 0 \text{ dB}, B_{fs} = 0.4, u_I = 0.29, INR = 30 \text{ dB}, B_{fI} = 0.4, LNR = 0 \text{ dB}, 5 \text{ dB}, \text{ and } -50 \text{ dB}$ , 100 trials; average  $SINR_o$  versus  $K$ .

### 7.11.2 LMS Implementation

In this section, we discuss the LMS implementation of the TDL beamformer. In the first subsection we use the direct form structure in Figure 7.81. In the second subsection we use the GSC structure in Figure 7.82.

#### 7.11.2.1 Direct-form FIR structure

We modify the results in Section 7.7 to accommodate the FIR case. The starting point is (7.418) – (7.420). Recalling that

$$\mathbf{w}_q = \mathbf{C} \left( \mathbf{C}^H \mathbf{C} \right)^{-1} \mathbf{g}, \quad (7.545)$$

and

$$\mathbf{P}_c^\perp \triangleq \mathbf{I} - \mathbf{C} \left( \mathbf{C}^H \mathbf{C} \right)^{-1} \mathbf{C}^H, \quad (7.546)$$

the LMS equations are given by (7.418) – (7.420). For the FIR model,

$$\mathbf{w}(0) = \mathbf{w}_q, \quad (7.547)$$

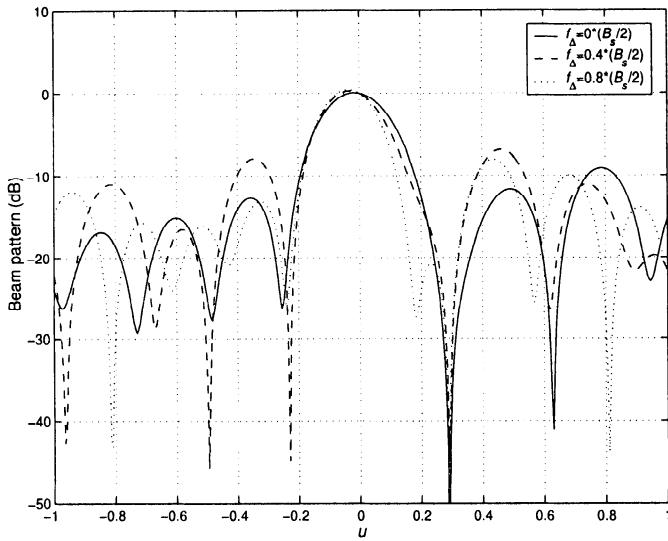


Figure 7.84 Broadband MPDR time-domain beamformer with SMI algorithm:  $M = 4, u_s = 0, SNR = 0 \text{ dB}, B_{fs} = 0.4, u_I = 0.29, INR = 30 \text{ dB}, B_{fI} = 0.4, LNR = 0 \text{ dB}, 5 \text{ dB}, \text{ and } -50 \text{ dB}, 100 \text{ trials}$ ; representative beam patterns at  $K = 1,000$  at three frequencies.

and

$$\mathbf{w}(K) = \mathbf{P}_c^\perp [\hat{\mathbf{w}}(K-1) - \Delta(K)y_p(K)\mathbf{x}(K)] + \mathbf{w}_q, \quad (7.548)$$

where

$$y_p(K) = \mathbf{x}^H(K)\mathbf{w}(K-1), \quad (7.549)$$

and

$$\Delta(k) = \text{diag}[\Delta_0(k), \Delta_1(k), \dots, \Delta_m(k)] \otimes \mathbf{I}_N. \quad (7.550)$$

We first consider the MPDR case in which (from (6.797))

$$\mathbf{C} = \begin{bmatrix} \mathbf{1} & & & \\ & \mathbf{1} & & \\ & & \ddots & \\ & & & \mathbf{1} \end{bmatrix} = \mathbf{I}_M \otimes \mathbf{1}_N, \quad (7.551)$$

and

$$\mathbf{g} = [1 \ 0 \ \dots \ 0]^T. \quad (7.552)$$

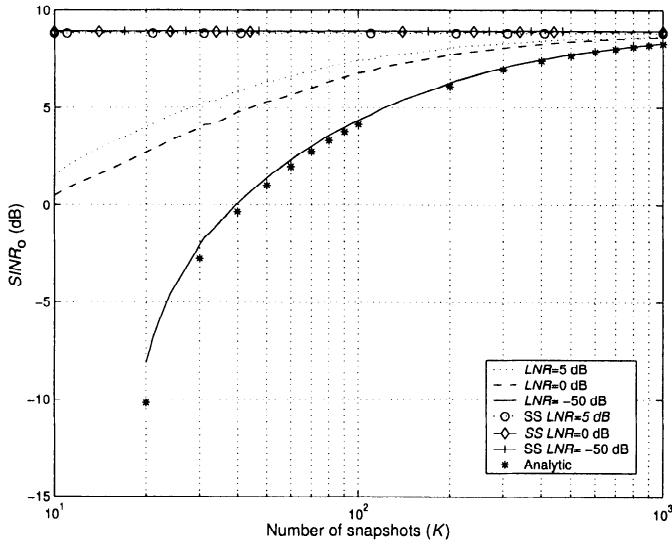


Figure 7.85 Broadband MPDR time-domain beamformer with SMI algorithm:  $M = 2$ ,  $u_s = 0$ ,  $SNR = 0$  dB,  $B_{fs} = 0.4$ ,  $u_I = 0.29$ ,  $INR = 30$  dB,  $B_{fI} = 0.4$ ,  $LNR = 0$  dB, and  $-50$  dB, 100 trials; average  $SINR_o$  versus  $K$ ; steady state results are also shown.

Using (7.551) in (7.546), we have

$$\mathbf{P}_c^\perp = \mathbf{I}_{NM} - \frac{1}{N} \left( \mathbf{I}_M \otimes \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \ddots & & \vdots \\ 1 & & & 1 \end{bmatrix} \right). \quad (7.553)$$

Using (7.552) and (7.553) in (7.548) gives the LMS equations,

$$\begin{aligned} \mathbf{w}_0(K) &= \mathbf{w}_0(K-1) - \Delta_0(K) y_p(K) \mathbf{x}_0(K) \\ &\quad - \frac{1}{N} \mathbf{1} \mathbf{1}^T \times [\mathbf{w}_0(K-1) - \Delta_0(K) y_p(K) \mathbf{x}_0(K)] + \frac{1}{N} \mathbf{1}, \end{aligned} \quad (7.554)$$

$$\begin{aligned} \mathbf{w}_m(K) &= \mathbf{w}_m(K-1) - \Delta_m(K) y_p(K) \mathbf{x}_m(K) \\ &\quad - \frac{1}{N} \mathbf{1}^T [\mathbf{w}_m(K-1) - \Delta_m(K) y_p(K) \mathbf{x}_m(K)], \\ m &= 1, \dots, M-1, \end{aligned} \quad (7.555)$$

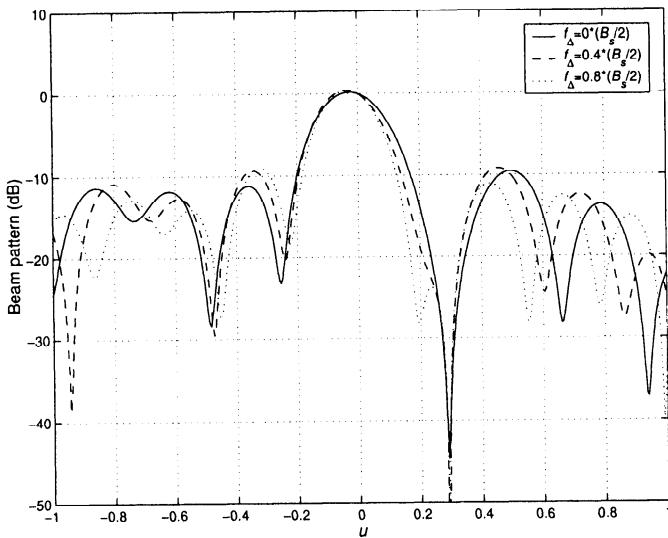


Figure 7.86 Broadband MPDR time-domain beamformer with SMI algorithm:  $M = 2$ ,  $u_s = 0$ ,  $SNR = 0$  dB,  $B_{fs} = 0.4$ ,  $u_I = 0.29$ ,  $INR = 30$  dB,  $B_{fI} = 0.4$ ,  $LNR = 0$  dB, 100 trials; representative beam patterns at  $K = 1,000$  at three frequencies.

and

$$y_p(K) = \sum_{m=0}^{M-1} \mathbf{x}_m^H(K) \mathbf{w}_m(K-1). \quad (7.556)$$

Note that the coupling of the tap weight vectors only occurs in (7.556). This result is because of the diagonal structure of  $\Delta(k)$  in (7.550). Therefore, we can use different  $\Delta_m(K)$  in (7.554)–(7.555). We used the power normalized version,

$$\Delta_m(K) = \frac{\gamma_m}{\sigma_m^2(K)}, \quad (7.557)$$

where

$$\sigma_m^2(K) = \beta \sigma_m^2(K-1) + (1-\beta) \| \mathbf{x}_m(K) \|^2 \quad (7.558)$$

with

$$\sigma_m^2(0) = 0, \quad m = 0, \dots, M-1. \quad (7.559)$$

The next example uses the same signal and interference model as Example 7.11.2.

#### Example 7.11.2 (continuation, Example 7.11.1)

Consider a uniform 10-element array with spacing  $d = \lambda_u/2$ . There is a 4-tap FIR filter at the output of each sensor's quadrature demodulator. The desired signal arrives

from broadside ( $u_s = 0$ ) with a flat spectrum with  $B_f = 0.4$ . The interfering signal arrives from  $u_I = 0.29$  with an identical spectrum. The  $SNR = 0$  dB and the  $INR = 30$  dB. We use the LMS algorithm in (7.554) – (7.559) with  $\gamma = 0.01$ ,  $m = 0, \dots, M-1$ , and  $\beta = 0.99$ . Two levels of  $LNR$ , 0 dB, and  $-50$  dB are considered.

In Figure 7.87, the average  $SINR_o$  is plotted versus  $K$ . The LMS algorithm converges much more slowly than the SMI algorithm. In Figure 7.88, representative beam patterns at  $K = 1000$  are plotted for various frequencies in the band. The advantage of LMS is the reduced computational complexity.

### 7.11.2.2 GSC FIR structure

The GSC implementation using an FIR filter is shown in Figure 7.82. We assume that the constraint matrix is block diagonal so that  $\mathbf{B}^H$  is a  $(N - N_d) \times N$  matrix. The output of the blocking matrix is an  $(N - N_d) \times 1$  vector,  $\mathbf{x}_b(k)$ . We define a  $(N - N_d)M \times 1$  vector,  $\mathbf{z}$ ,

$$\mathbf{z}(k) \triangleq \begin{bmatrix} \mathbf{x}_b(k) \\ \mathbf{x}_b(k - T_s) \\ \vdots \\ \mathbf{x}_b(k - (M-1)T_s) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x_{b,00}(k) \\ x_{b,10}(k) \\ \vdots \\ x_{b,00}(k - T_s) \\ x_{b,10}(k - T_s) \\ \vdots \\ x_{b,00}(k - (M-1)T_s) \\ x_{b,10}(k - (M-1)T_s) \\ \vdots \end{bmatrix} \end{bmatrix}. \quad (7.560)$$

The LMS equations are (from (7.425) – (7.427)),

$$e_{pa}(K) = y_c(K) - \hat{\mathbf{w}}_a^H(K-1)\mathbf{z}(K), \quad (7.561)$$

$$\hat{\mathbf{w}}_a(K) = \hat{\mathbf{w}}_a^H(K-1) + \Delta(K)\mathbf{z}(K)e_{pa}^*(K), \quad (7.562)$$

and

$$\hat{\mathbf{w}}(K) = \mathbf{w}_q - \mathbf{B}\hat{\mathbf{w}}_a(K). \quad (7.563)$$

The LMS equations are coupled only through (7.561).

$$\hat{w}_{a,00}(K) = \hat{w}_{a,00}^H(K-1) + \Delta_0(K)x_{b,00}(K)e_{pa}^*(K), \quad (7.564)$$

$\vdots$

$$\hat{w}_{a,rq}(K) = \hat{w}_{a,rq}^H(K-1) + \Delta_r(K)x_{b,rq}(K)e_{pa}^*(K)$$

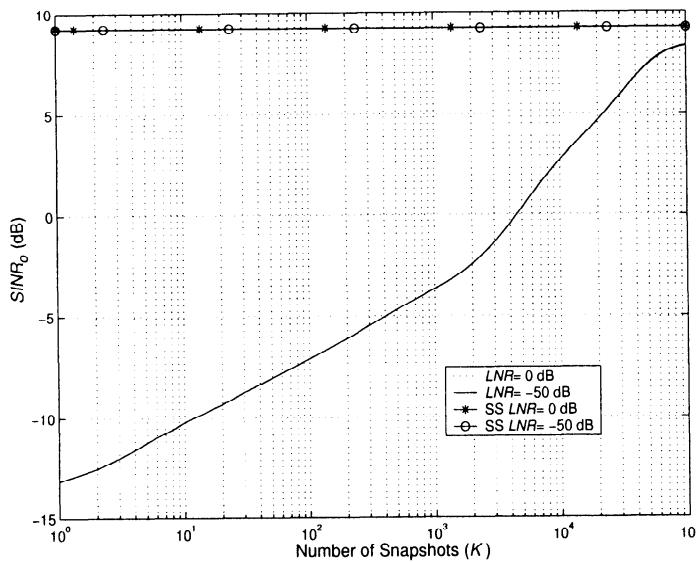


Figure 7.87 Broadband MPDR time-domain beamformer with LMS algorithm:  $M = 4, u_s = 0, SNR = 0 \text{ dB}, B_{fs} = 0.4, u_I = 0.29, INR = 30 \text{ dB}, B_{fi} = 0.4, LNR = 0 \text{ dB}$ , and  $-50 \text{ dB}$ , 100 trials; average  $SINR_o$  versus  $K$ .

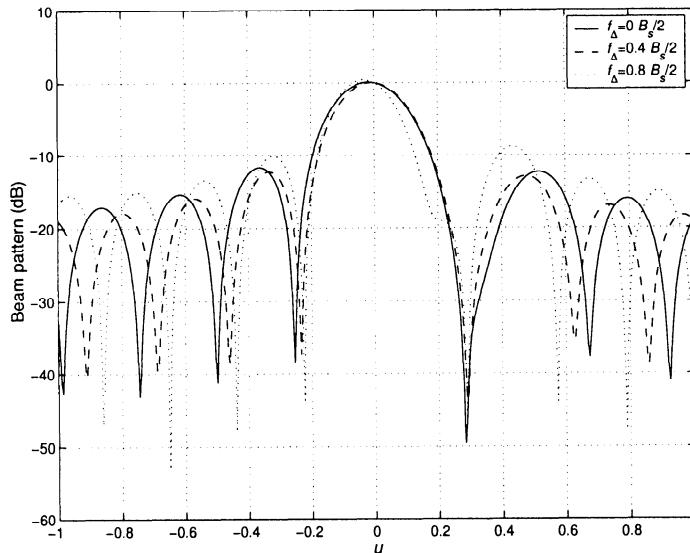


Figure 7.88 Broadband MPDR time-domain beamformer with LMS algorithm:  $M = 4, u_s = 0, SNR = 0 \text{ dB}, B_{fs} = 0.4, u_I = 0.29, INR = 30 \text{ dB}, B_{fi} = 0.4, LNR = 0 \text{ dB}$ , 100 trials; representative beam patterns at  $K = 1,000$  at three frequencies.

$$\begin{aligned}
&= \hat{w}_{a,rq}^H(K-1) + \Delta_r(K)x_{b,r0}(K-qT_s)e_{pa}^*(K), \\
&\quad r = 0, \dots, (N - N_d) - 1, \\
&\quad q = 0, \dots, M - 1.
\end{aligned} \tag{7.565}$$

Because it is an unconstrained optimization, the summation down the array in (7.555) is not required. The convergence parameter is a function of  $r$ , the index down the blocking matrix. We use the PNLMS algorithm,

$$\Delta_r(K) = \frac{\gamma_r}{\sigma_x^2(K)}, \tag{7.566}$$

where

$$\sigma_x^2(K) = \beta\sigma_x^2(K-1) + (1-\beta)\|\mathbf{x}_{b,q}\|^2. \tag{7.567}$$

The GSC implementation has the same transient behavior as the direct-form implementation if the same values of  $\gamma$  and  $\beta$  are used.

### 7.11.2.3 Summary

The FIR implementation using the LMS algorithm is one of the more widely used implementations of time-domain broadband beamformers. The advantage is that the computational complexity is the least of the various algorithms. The disadvantage is that the rate of convergence is a function of the eigenvalue spread and may be slow in some environments.

In the next section, we consider an alternative implementation of the FIR filter that improves the convergence behavior.

### 7.11.3 GSC: Multichannel Lattice Filters

In this section, we discuss an alternative implementation of the FIR filters in the bottom branch of the GSC. We implement the filtering in the bottom branch using a multichannel lattice filter. Single-channel lattice filters are widely used in adaptive filtering applications because of their convergence properties.

Early applications included speech analysis and synthesis, [IS71], linear prediction, [MV78], and least square estimation. The early papers by Griffiths [Gri77], [Gri78] are particularly relevant to our application. In the past twenty years there have been a large number of papers dealing with adaptive algorithms using lattice structures. Chapter 6 (by F. Ling) in [KT93], Chapters 4 – 7 in Proakis et al. [PRLN92], and Chapters 6 and 15 in [Hay96] provide thorough discussions. Most of the applications have

considered single-input systems. For the array processing problem we need a multichannel lattice filter implementation because the input is a  $N \times 1$  complex vector. Lattice filters are based on MMSE theory so we implement our processor as a GSC in which the desired scalar signal is  $y_c(k)$ .

The multichannel lattice filter implementation for the array processing problem is given in Griffiths [Gri77], [Gri78]. Youn and Chang [YC86] have discussed the implementation in more detail. A similar algorithm is given in Table 6.9 on p. 239 of [KT93].

The advantage of the lattice structure is that the equations to update the stages are uncoupled. Therefore, one can choose the adaption parameters in each stage in a manner that improves convergence. An LMS algorithm to update the stages is given in [Gri78] and [YC86].

Just as in the narrowband case, we can improve the adaptive behavior by using an RLS algorithm. QRD multichannel lattice algorithms are discussed by McWhirter and Proudler in Section 7.4.2 of [KT93]. They are also discussed by Ling in Section 6.3.3 of [KT93]. The reference lists at the end of these two chapters provide further material. Both Haykin [Hay96] and Proakis et.al [PRLN92] contain discussions. The interested reader is referred to these references for a complete discussion.<sup>35</sup>

#### 7.11.4 Summary

In this section we have provided a brief discussion of the adaptive behavior of time-domain broadband beamformers. We restricted our discussion to SMI and LMS algorithms.

### 7.12 Summary

In this chapter, we have studied adaptive beamformers. The topics in the chapter can be divided into four parts.

The first part of the chapter (Section 7.2) developed techniques for estimating the spatial spectral matrix. We emphasized the sample spectral matrix and the FB averaged sample spectral matrix because of their maximum likelihood character, their relative simplicity, and their widespread usage. Other techniques that added more structure to the model were introduced but not developed.

---

<sup>35</sup>Our discussion of lattice filters requires the reader to consult the suggested references to understand the issues. A more self contained discussion would require a lot of additional background that we have not developed.

The second part of the chapter (Sections 7.3–7.7) developed the three adaptive algorithms that are most widely used.

The SMI algorithm is a block processing algorithm in which the estimated spatial spectral matrix is substituted for the ensemble spectral matrix. We found that we needed to include diagonal loading to obtain satisfactory performance. The convergence to steady state is a function of  $K/N$ . The disadvantage of the SMI approach is that the computational complexity is a function of  $N^3$ , so that the SMI algorithm is only practical for modest size arrays.

The RLS algorithm can be developed in several different ways. We used a least squares formulation and then developed a recursive implementation that was similar to a Kalman filter. A recursive QRD version was developed that was stable and computationally efficient. Once again, diagonal loading was utilized.

The LMS algorithm required the least amount of computation. We developed it as a stochastic version of the steepest descent algorithm. The disadvantage of the LMS algorithm was that its convergence depended on the eigenvalue spread and may be slow in certain environments.

The third part of the chapter (Sections 7.8–7.10) focused on techniques for reducing the dimension of the adaptive processor. The two techniques were eigenspace processing and beamspace processor. In order to implement eigenspace processors, we must estimate the dimension of the signal subspace. In Section 7.8, we developed the MDL and AIC algorithms to estimate the signal subspace dimension and briefly discussed subspace tracking algorithms. We found that eigenspace beamformers were very effective and reduced the convergence time and computational complexity. The disadvantage of eigenspace beamformers is that, under certain scenarios, we may lose a component of the eigenspace that is highly correlated with the signal. This loss causes a dramatic degradation in performance. BS processors construct the subspace in a deterministic manner, normally with a set of orthogonal beams spaced around the steering direction. We then utilize the SMI, RLS, or LMS algorithm in beamspace. The resulting algorithms are computationally simpler and converge faster than the corresponding element space algorithms.

The fourth part of the chapter (Section 7.11) discussed adaptive time domain implementations of broadband beamformers. The discussion introduced the issues and considered several examples of LMS and SMI implementations. We referred the reader to the literature for a discussion of lattice structures and RLS implementations.

There are several issues that we introduced in Chapter 6 that we have

not discussed in the adaptive context:

- (i) **Array perturbations.** In Section 6.6.3, we discussed how array perturbations degrade the steady state performance. Similar effects occur in the adaptive case, but no new issues arise. Wax and Anu [WA96] discuss the closely related problems of steering vector errors and finite data.
- (ii) **Coherent signals and interferences.** In Section 6.12, we discussed beamforming when the desired signals were coherent or correlated. We showed that spatial smoothing provided an effective algorithm in many scenarios. The adaptive version of the algorithm is developed in Shan and Kailath [SK85] and Takao and Kikuma [TK87]. The reader is referred to these references for a discussion of adaptive spatial smoothing.
- (iii) **Covariance augmentation.** In Section 6.7.6, we discussed the use of covariance matrix tapers to broaden the nulls created by the adaptive beamformer. A discussion of the adaptive version is given in [Gue99], [Gue00], [Zat99] and [Zat00]. Another technique for broadening the nulls by imposing derivative constraints is discussed in [GSB96].

This chapter completes our discussion of beamforming. In the next two chapters we discuss the parameter estimation problem.

## 7.13 Problems

### P7.2 Estimation of Spatial Spectral Matrices

#### Problem 7.2.1

Compute the bias of the sample spectral matrix as an estimator of  $\mathbf{S}_x$ .

#### Problem 7.2.2 [Wil62]

(a) Show that the chi-squared density is reproductive with respect to  $K/2$ . In other words,

$$E [\chi^{2r}(K)] = \frac{2^r \Gamma\left(\frac{K}{2} + r\right)}{\Gamma\left(\frac{K}{2}\right)}. \quad (7.568)$$

(b) Using the result in part (a), show that

$$E [\chi^2(K)] = K, \quad (7.569)$$

$$E [\chi^4(K)] = K(K+2), \quad (7.570)$$

$$\text{var} [\chi^2(K)] = K(K+2) - K^2 = 2K. \quad (7.571)$$

**Problem 7.2.3**

Show that

$$\begin{aligned} E\left[\frac{1}{\chi^2(K)}\right] &= \frac{1}{2} \frac{\Gamma\left(\frac{K}{2} - 1\right)}{\Gamma\left(\frac{K}{2}\right)} \\ &= \frac{1}{2} \frac{\Gamma\left(\frac{K}{2} - 1\right)}{\left(\frac{K}{2} - 1\right) \Gamma\left(\frac{K}{2} - 1\right)} = \frac{1}{K - 2}. \end{aligned} \quad (7.572)$$

**Problem 7.2.4**

Extend the technique of FB averaging to a standard linear array with an odd number of elements  $N$ .

**Problem 7.2.5**

Consider a standard 19-element hexagonal array. Show how to construct a conjugate symmetric array manifold vector.

**Problem 7.2.6**

Consider the correlation matrix  $\mathbf{S}_x$  in (7.75). Assume that

$$\mathbf{S}_f = \begin{bmatrix} 1 & e^{j\alpha\pi} \\ e^{-j\alpha\pi} & 1 \end{bmatrix}. \quad (7.573)$$

Assume that FB averaging is used.

- (a) Find  $\mathbf{S}_{x,fb}$ .
- (b) Discuss the behavior as a function of  $\alpha$ .

**Problem 7.2.7**

Generate a Gaussian random sequence corresponding to a real AR(1) process acting as an input to a standard 10-element linear array.

- (a) Calculate  $\hat{\mathbf{S}}_x$  and  $\hat{\mathbf{S}}_{x,fb}$  for  $K = 10, 20, 100, 1000$  for one trial.
- (b) Repeat for 50 trials and  $K = 100$  and analyze your results.

**Problem 7.2.8**

In this problem we find the least squares estimate of  $\mathbf{S}_x$  with a Toeplitz constraint imposed.<sup>36</sup> Specifically, we define

$$e^2 = \text{tr} \left[ (\mathbf{C}_x - \mathbf{S}_x)^T (\mathbf{C}_x - \mathbf{S}_x) \right], \quad (7.574)$$

where  $\mathbf{C}_x$  is the sample covariance matrix defined in (7.3). The squared error is the Frobenius norm,

$$e^2 = \sum_{i=1}^N \sum_{j=1}^N (c_{ij} - s_{ij})^2. \quad (7.575)$$

---

<sup>36</sup>Our discussion follows Scharf [Sch91], which in turn references Lawson and Hansen [LH74].

- (a) We impose the structure on  $\mathbf{S}_x$  through the vector parameter  $\theta$  and denote it by  $\mathbf{S}_x(\theta)$ . Minimize  $e^2$  with respect to  $\theta$  and show that

$$\frac{\partial e^2}{\partial \theta_n} = -2 \operatorname{tr} [(\mathbf{C}_x - \mathbf{S}_x)]|_{\mathbf{S}_x=\hat{\mathbf{S}}_x} = 0. \quad (7.576)$$

- (b) Show that the second derivative is

$$\operatorname{tr} \left[ \left( \frac{\partial \mathbf{S}_x}{\partial \theta_n} \right)^T \frac{\partial \mathbf{S}_x}{\partial \theta_n} \right] + \operatorname{tr} \left[ (\mathbf{C}_x - \mathbf{S}_x)^T \frac{\partial^2 \mathbf{S}_x}{\partial \theta_n} \right], \quad (7.577)$$

which will be evaluated to verify that the solution is a minimum.

- (c) For a Hermitian Toeplitz structure we parameterize  $\mathbf{S}_x(\theta)$  as

$$\mathbf{S}_x = s_0 \mathbf{Q}_0 + \sum_{n=1}^{N-1} s_n \mathbf{Q}_n + \sum_{n=1}^N s_n^* \mathbf{Q}_n^T. \quad (7.578)$$

This corresponds to,

$$\mathbf{S}_x = \begin{bmatrix} s_0 & s_1 & \cdots & s_{N-1} \\ s_1^* & s_0 & \ddots & \\ \vdots & \ddots & s_1 & \\ s_{N-1} & \cdots & s_1^* & s_0 \end{bmatrix}, \quad (7.579)$$

where

$$\mathbf{Q}_0 = \mathbf{I}. \quad (7.580)$$

and

$$\mathbf{Q}_n = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \\ 0 & & 1 & & \\ & \ddots & & & \\ & & & & 0 \end{bmatrix}. \quad (7.581)$$

is an upper triangular matrix with 1s along the  $n$ th diagonal and zeros everywhere. Differentiate with respect to  $s_0$ ,  $\operatorname{Re}[s_n]$ , and  $\operatorname{Im}[s_n]$  and set the result to zero.

$$\operatorname{tr} [(\mathbf{C}_x - \mathbf{S}_x)^T \mathbf{I}] = 0. \quad (7.582)$$

$$\operatorname{tr} [(\mathbf{C}_x - \mathbf{S}_x)^T (\mathbf{Q}_n + \mathbf{Q}_n^T)] = 0. \quad (7.583)$$

$$\text{tr} \left[ (\mathbf{C}_x - \mathbf{S}_x)^T (\mathbf{Q}_n - \mathbf{Q}_n^T) \right] = 0. \quad (7.584)$$

(d) Use the above results and show that,

$$\hat{s}_0 = \frac{1}{N} \sum_{n=1}^N \mathbf{C}_{ii}, \quad (7.585)$$

$$Re [\hat{s}_n] = \frac{1}{2(N-M)} \left[ \sum_{j=n+1}^N \mathbf{C}_{(j-n)j} + \sum_{i=n+1}^N \mathbf{C}_{i(i-n)} \right], \quad (7.586)$$

and

$$Im [\hat{s}_n] = \frac{j}{2(N-n)} \left[ \sum_{j=n+1}^N \mathbf{C}_{(j-n)j} - \sum_{i=n+1}^N \mathbf{C}_{i(i-n)} \right], \quad (7.587)$$

We see that the solution averages along the appropriate diagonals to obtain a Hermitian Toeplitz matrix. We refer to this estimated matrix as  $\hat{\mathbf{S}}_{x,Toe}$ . This estimate is intuitively satisfying. However, the resulting estimate is not non-negative definite and gives poor results in some applications.

### Problem 7.2.9

Consider the model in which  $\mathbf{x}(k)$  is generated by two equal power, uncorrelated plane waves plus white noise.

(a) Find an expression for the asymptotic (large  $K$ ) behavior of the eigenvalues and eigenvectors.

(b) Plot for a standard 10-element linear array.

## P7.3 Sample Matrix Inversion (SMI)

**Problem Note 7.3.1:** In this section, we simulate the performance of the SMI implementation of adaptive beamformers for various scenarios including those in Table 7.4. The scenarios are chosen to be representative of problems that may be encountered in practical applications. In subsequent sections, we consider different adaptive implementations for the same scenarios. The same data set should be used for the different beamformers. Use 100 trials in your simulation. All of the scenarios include additive white noise.

### Problem 7.3.1

Consider a SLA10 and Test Scenario 1.

- (a) Simulate the  $\mathbf{w}_{mpdr,smi}^H$  beamformer in (7.85). Choose several values of  $SNR$  and  $INR$ . Plot  $SINR_o$  versus  $K$ . Plot representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ . Discuss your results.
- (b) Repeat part (a) with various levels of fixed diagonal loading. Discuss your results.

### Problem 7.3.2 (continuation)

Repeat Problem 7.3.1 for Test Scenario 1m with  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.3** (continuation, Problem 7.3.1)

Consider the model in Problems 7.3.1. Implement  $\mathbf{w}_{mpdr,smi}^H$  as a GSC with  $\mathbf{B}^H \mathbf{B} = \mathbf{I}$ . Repeat Problem 7.3.1. Compare the performance on specific sample functions as well as an averaged basis.

Table 7.4

| Test Scenario | Signal                                   | Interferers  |  |
|---------------|--|--|--|
|               |  | Location   | Strength   |
| 1             | $u_m = 0$<br>$SNR = 0, 10, \dots, 30$ dB | Separated Sidelobe<br>$u_1 = 0.30$<br>$u_2 = 0.50$<br>$u_3 = 0.70$<br>$INR = 0, \dots, 40$ dB                  | $INR_i = 0, \dots, 40$ dB<br>i=1, 2, 3                 |
| 1m            | $ u_a  \leq u_m$                         | TS1  | TS1  |
| 2             | $u_s = 0$<br>$SNR = 0, \dots, 30$ dB     | Clustered Sidelobe<br>$u_1 = 0.28$<br>$u_2 = 0.30$<br>$u_3 = 0.32$   | Strength<br>$INR_i = 0, \dots, 40$ dB<br>i=1, 2, 3     |
| 2m            | $ u_a  \leq u_m$                         | TS2  | TS2  |
| 3             | $u_s = 0$<br>$SNR = 0, \dots, 30$ dB     | Outer Main Lobe<br>$u_1 = 0.18$<br>$u_2 = -0.30$   | Strength<br>$INR_i = 0, \dots, 40$ dB<br>i=1, 2        |
| 3a            | $ u_a  \leq u_m$                         | TS3  | TS3  |
| 4             | $u_s = 0$                                | Main Lobe<br>$u_1 = 0.08$  | Strength<br>$INR_i = 0, \dots, 40$ dB                  |
| 5             | $u_s = 0$                                | Random Sidelobe<br>$ u_i  \geq 0.20$<br>$u_i = 1, \dots, D$  | Strength<br>$INR_i = 0, \dots, 40$ dB<br>i = 0, ..., D |
| 6             | $u_s = 0$                                | Correlated Interference<br>$u_1 = 0.30$<br>$ \rho  \neq 0$   | $INR = 0, \dots, 40$ dB                                |
| 7             | $u_s = 0$                                | Spatially Spread<br>Section 5.7.1<br>Complex AR(1)<br>$\phi_a = 0.3,  a(1)  = 0.5,$<br>$0.7, \text{ and } 0.9$ | $INR = 10, 20, \text{ and } 30$ dB                     |

## Notes for Table 7.4

- In TS5, assume that the location of each interferer is a uniform random variable  $0.20 \leq |u_i| \leq 0.90$ ,  $i = 1, \dots, D$  and that the locations are statistically independent. The  $INR_i$  takes on one of five values, 0, 10, 20, 30, and 40 dB with equal probability. The locations and  $INRs$  of the interferers are statistically independent. They are fixed over the snapshots but are different realizations on each trial.
- In TS7, the additive white noise causes the interference plus noise model to be an ARMA process.
- In TS3, an extra interferer is added in the sidelobe.

**Problem 7.3.4**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 2.

**Problem 7.3.5**

Repeat Problem 7.3.2 for a SLA10 and Test Scenario 2m.

**Problem 7.3.6**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 3.

**Problem 7.3.7**

Repeat Problem 7.3.2 for a SLA10 and Test Scenario 3m.

**Problem 7.3.8**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 4.

**Problem 7.3.9**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 5 with  $D = 4$ .

**Problem 7.3.10**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 6 with  $\rho = 0.95 \exp(j\pi/4)$ .

**Problem 7.3.11**

Repeat Problem 7.3.1 for a SLA10 and Test Scenario 7.

**Problem Note 7.3.2:** In several of the previous problems, the scenario included signal mismatch or a main-lobe interferer. We attempted to provide main-lobe protection by using diagonal loading. In Section 6.7, we utilized linear constraints to provide main-lobe protection. The linear constraint sets included:

- (a) Directional (6.287).
- (b) Derivative (6.308), (6.310).
- (c) Eigenvector (6.344).
- (d) Quiescent pattern (6.419).

In the next five problems, we revisit some of the previous problems where main-lobe protection was needed. In each problem, choose one or more of the above constraints and implement the LCMP-SMI beamformer. Choose several values of  $SNR$  and  $INR$  and use various  $LNR$ . In each problem,

- (a) Plot  $SINR_o$  versus  $K$ .
- (b) Plot representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

Compare the diagonal loading level needed in the LCMP beamformer with the diagonal loading level needed in the MPDR beamformer.

**Problem 7.3.12** (continuation, Problem 7.3.2)

Repeat Problem 7.3.2 for a SLA10 and Test Scenario 1m. Use  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.13** (continuation, Problem 7.3.5)

Repeat Problem 7.3.5 for a SLA10 and Test Scenario 2m. Use  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.14** (continuation, Problem 7.3.7)

Repeat Problem 7.3.7 for a SLA10 and Test Scenario 3m. Use  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.15** (continuation, Problem 7.3.8)

Repeat Problem 7.3.8 for a SLA10 and Test Scenario 4m. Use  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.16** (continuation, Problem 7.3.10)

Repeat Problem 7.3.10 for a SLA10 and Test Scenario 6m. Use  $u_a = 0.02$  and  $0.04$ .

**Problem 7.3.17**

Consider a SLA20. Assume  $u_s = 0$  and the *SNR* is 10 dB. Assume there are two interfering signals at  $u_{I1} = 0.15$  and  $u_{I2} = -0.45$ . Each interfering signal has an *INR* = 30 dB.

The random variable  $\rho$  is defined in (7.92) and the random variable  $\eta$  is defined in (7.106).

- (a) Plot  $E[\rho]$  and  $\bar{\rho}$  for  $K = 20, 40, 60, 80$ , and  $100$  for MVDR beamformer using  $\hat{\mathbf{S}}_n = \mathbf{C}_n$ : use 200 trials.
- (b) Repeat for the case of  $\hat{\mathbf{S}}_n = \mathbf{C}_{n,fb}$ . Plot  $\bar{\rho}$ . Also plot  $E[\rho]$  from (7.95) replacing  $K$  with  $2K$ .
- (c) Plot a histogram of  $\rho$  for  $K = 40$ . Use 500 trials.
- (d) Consider the case when  $K = 4, 10$ , and  $16$ . Add DL (perhaps *LNR* = 10 dB) and plot  $\rho$ .
- (e) Plot  $\bar{\eta}$  for the MPDR beamformer for  $K = 20, 40, 60, 80$ , and  $100$ . Compare to the analytic expression given by (7.110) and (7.111).
- (f) Repeat parts (a) and (b) for the GSC configuration. Specify the  $\mathbf{B}$  matrix that you use ( $\mathbf{B}^H \mathbf{B} = \mathbf{I}$ ).

**P7.4 Recursive Least Squares**

**Problem Note 7.4.1:** In this section, we simulate the performance of the RLS implementation for various scenarios including those in Table 7.4. The parameters that must be specified are  $\mu$  in (7.138), the initial conditions ( $\sigma_o^2$  in (7.161) and  $\hat{\mathbf{w}}(0)$  in (7.163)), and the diagonal loading  $\sigma_L^2$  in (7.209). In each problem, consider four RLS implementations:

- (a) RLS with no diagonal loading.
- (b) GSC-RLS with fixed diagonal loading (Section 7.4.4).  $\sigma_L^2/\sigma_w^2$  must be specified. (Include  $\sigma_L^2 = 0$  as a reference).
- (c) GSC-RLS with variable diagonal loading (Section 7.4.5).  $T_o$  must be specified.
- (d) Conjugate symmetric RLS (Section 7.4.6) (in order to use diagonal loading, we must derive the GSC implementation of RLS-FB; see Problem 7.4.12).

You must choose the values of these parameters in your solution. Use the same data set as in Section P.7.3. In each problem,

- (a) Plot  $SINR_o$  versus  $K$ .
- (b) Plot representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

Compare your results to SMI results for the same scenario. Discuss the effects of the various parameter values. Discuss any computational issues. Conduct 100 trials.

The test scenario from Table 7.4 that we use with each problem is shown in Table 7.5. Recall that the details of the scenario were given in the SMI problem section. We also show the problem numbers for P7.5, P7.6, and P7.7.

**Problem Note 7.4.2:** The comments in Problem Note 7.3.2 also apply to the RLS implementation. In the next five problems we consider LCMP beamformers implemented in a GSC structure. Consider one or more of the constraint sets in Problem Note 7.3.2 and plot the results indicated in Problem Note 7.3.2. The problems are listed in Table 7.5.

Table 7.5

| Test Scenario | SMI    | RLS    | QRD    | SD     | LMS    |
|---------------|--------|--------|--------|--------|--------|
| 1             | 7.3.1  | 7.4.1  | 7.5.1  | 7.6.5  | 7.7.1  |
| 1m            | 7.3.2  | 7.4.2  | 7.5.2  | 7.6.6  | 7.7.2  |
| 2             | 7.3.4  | 7.4.3  | 7.5.3  | 7.6.7  | 7.7.3  |
| 2m            | 7.3.5  | 7.4.4  | 7.5.4  | 7.6.8  | 7.7.4  |
| 3             | 7.3.6  | 7.4.5  | 7.5.5  | 7.6.9  | 7.7.5  |
| 3m            | 7.3.7  | 7.4.6  | 7.5.6  | 7.6.10 | 7.7.6  |
| 4             | 7.3.8  | 7.4.7  | 7.5.7  | 7.6.11 | 7.7.7  |
| 5             | 7.3.9  | 7.4.8  | 7.5.8  | 7.6.12 | 7.7.8  |
| 6             | 7.3.10 | 7.4.9  | 7.5.9  | 7.6.13 | 7.7.9  |
| 7             | 7.3.11 | 7.4.10 | 7.5.10 | 7.6.14 | 7.7.10 |
| 1m            | 7.3.12 | 7.4.11 | 7.5.11 | 7.6.15 | 7.7.11 |
| 2m            | 7.3.13 | 7.4.12 | 7.5.12 | 7.6.16 | 7.7.12 |
| 3m            | 7.3.14 | 7.4.13 | 7.5.13 | 7.6.17 | 7.7.13 |
| 4             | 7.3.15 | 7.4.14 | 7.5.14 | 7.6.18 | 7.7.14 |
| 6             | 7.3.16 | 7.4.15 | 7.5.15 | 7.6.19 | 7.7.15 |

### Problem 7.4.16

- (a) Derive the equations that specify the GSC implementation of the conjugate symmetric RLS beamformer in Section 7.4.6. Your equations should use real computation.
- (b) Modify your results in part (a) to incorporate diagonal loading.

### Problem 7.4.17

Repeat Example 7.4.1 for the case in which there are two interferers at  $u_I = \pm 0.29$ . All other parameters remain the same. Compare your results to those in Problem 6.3.7.

### Problem 7.4.18

Consider the LCMP beamformer in Example 7.3.1. Develop a recursive GSC implementation of the beamformer. Simulate its performance and compare your results to those in Example 7.3.1.

### Problem 7.4.19

One of the reasons for using  $\mu \neq 1$  is to accommodate non-stationary environments. Consider the model in Problem 7.4.17. Denote the *INR* of the interferer at  $u_I = 0.29$  on the  $k$ th snapshot as  $INR_1(k)$ . Denote the *INR* of the interferer at  $u_I = -0.29$  on the  $k$ th snapshot as  $INR_2(k)$ . Assume

$$INR_1(k) = \begin{cases} 10^3, & k = 1, \dots, 50 \\ 0, & k = 51, \dots, 100 \\ 10^3, & k = 101, \dots, 150, \end{cases}$$

and continues in the same pattern.

$$INR_2(k) = \begin{cases} 10^3, & k = 1, \dots, 25 \\ 0, & k = 26, \dots, 75 \\ 10^3, & k = 76, \dots, 125, \end{cases}$$

and continues in the same pattern.

Study the behavior of the RLS MPDR beamformer for various choices of  $\mu$ . Use 10 dB diagonal loading and  $SNR = 10$  dB.

#### Problem 7.4.20

Repeat Example 7.4.1 for the scenario in Problem 7.3.17, that is,  $N = 20$ ,  $u_s = 0$ ,  $u_{I1} = 0.15$  and  $u_{I2} = -0.45$ . Consider the following parameters:

- (i)  $\mu = 0.95, 0.99, 0.999$ ,
- (ii)  $\sigma^2 = 10$ ,  $\sigma_{L,smi}^2 = 0$ . (i.e., no SMI DL),
- (iii)  $w(0) = v_s/N$ ,
- (iv)  $INR = 20$  dB,
- (v)  $SNR = 0, 10, 20$  dB.

Plot average  $SINR$  behavior as well as sample beampatterns.

#### Problem 7.4.21 (continuation)

Consider the same scenario as in Problem 7.4.20. Implement the FB RLS beamformer in Section 7.4.6. Compare your results to those in Problem 7.4.20. Comment on the difference in average  $SINR$ , beampatterns, and computational load.

#### Problem 7.4.22 (continuation)

Consider the same scenario as in Problem 7.4.20. Consider an LCMP beamformer with three mainbeam constraints. You may use any type of constraints you wish (e.g., directional, derivative, quiescent pattern), but specify your choice. Implement the direct form and GSC form of the LCMP beamformer. Assume  $u_s = 0.03$ . Compare your results to those obtained in Problems 7.4.20 and 7.4.21. Compare/comment on the computational complexity of the direct versus GSC beamformers.

#### Problem 7.4.23 (continuation)

Implement the GSC LCMP beamformer from Problem 7.4.22 with (1) fixed loading at 10 dB use (7.211), (2) variable loading, (3)scaled projection. Compare your results to those obtained in Problem 7.4.23. Use a constraint level of  $T_o = 2/N$ .

#### Problem 7.4.24

Repeat the derivation in (7.178)–(7.183) for the case of a moving window of length  $1/1 - \mu$ . Simulate its performance for several of the examples in the text.

## P7.5 Efficient Recursive Implementation Algorithms

**Problem Note 7.5.1:** The QRD that we have developed utilizes a GSC implementation of the RLS beamformer and does not incorporate diagonal loading. The objective is to obtain an algorithm that has good numerical stability and reduced computation. The first ten problems repeat implementation (b) in Problem Note 7.4.1, with  $\sigma_L^2 = 0$  for the

scenarios in Problem 7.4.1–7.4.10. In addition to understanding the algorithm, your goal should be to notice the improved numerical stability.

#### Problem 7.5.1

Consider a SLA10 and Test Scenario 1. Implement the RLS beamformer using a GSC configuration. Implement the adaptive component in the lower path using a QRD.

- (a) Plot  $SINR_o$  versus  $K$ .
- (b) Plot representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

The remaining problem numbers are shown in Table 7.5 (Problems 7.5.2–7.5.10).

#### Problem Note 7.5.2:

The comments in Problem Notes 7.3.2 and 7.4.2 also apply to the QRD implementation. In the next five problems we consider the same models as in Problems 7.4.11–7.4.15 and use a GSC-RLS-QRD implementation. The problem numbers are shown in Table 7.5 (Problems 7.5.11–7.5.15).

#### Problem 7.5.16 (continuation, Problem 7.4.20)

Consider the same model as in Problem 7.4.20.

- (a) Implement the QRD algorithm for the RLS-GSC beamformer. Compare your results to those in Problem 7.4.20.
- (b) Implement the QRD algorithm for the direct form RLS beamformer. Compare your results to those in part (a).

## P7.6 Gradient Algorithms

#### Problem Note 7.6.1

The first set of problems consider direct form MMSE beamformers.

#### Problem 7.6.1

Consider the model in Example 7.6.5. Assume that the  $SNR = 0$  dB,  $INR_1 = 10$  dB,  $INR_2 = 50$  dB. Repeat the calculations in Example 7.6.5. Discuss your results.

#### Problem 7.6.2

Consider the model in Example 7.6.5. Assume that  $\mathbf{w}(0)$  is a Dolph-Chebychev pattern with  $-20$  dB sidelobes.

- (a) Repeat the calculations in Example 7.6.5.
- (b) Are  $v_n(K), n \geq 3$  equal to zero? If not, plot them as a function of  $K$ .

#### Problem 7.6.3

Consider a SRA10. The desired signal arrives at  $u_s = 0$  with an  $SNR = 10$  dB.

Assume that two interferers arrive at  $u_1$  and  $u_2$ . Consider the following parameters:

$$INR_1 = 30 \text{ dB}, u_1 = 0.29,$$

$$INR_2 = 45 \text{ dB}, u_2 = 0.49.$$

- (a) Repeat the calculations in Example 7.6.4 (choose an appropriate  $\alpha$ ). Discuss your results.

- (b) Modify the algorithm to include a variable step size  $\alpha(K)$ . Compare your results to those in part(a).

**Problem 7.6.4**

Repeat Problem 7.6.3 for the following parameter set:

$$INR_1 = 30 \text{ dB}, u_1 = 0.27,$$

$$INR_2 = 30 \text{ dB}, u_2 = 0.31.$$

**Problem Note 7.6.2**

The next set of problems utilize the test scenarios in Table 7.4. In all of the problems we use a GSC implementation and use the MMSE steepest descent algorithm to find  $\mathbf{w}_a$  in the lower branch ((7.387)–(7.389)). We assume that (7.385) is satisfied. In all problems, plot

- (a) Plot  $SINR_o$  versus  $K$ .
- (b) Plot representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

Compare your results to the RLS results in Problem Section 7.4. Note that we are using known ensemble statistics in this section. In Sections 7.3–7.5, we used measured statistics. A fairer comparison will be used when we simulate the LMS algorithm in Section 7.7.

The problem numbers are shown in Table 7.5.

## P7.7 LMS Algorithms

**Problem Note 7.7.1**

The first set of problems consider the test scenarios in Table 7.4. Consider LMS algorithms:

- (a) Griffiths LMS algorithm with diagonal loading (7.417).
- (b) Frost MPDR LMS algorithm, direct form with diagonal loading (7.418).
- (c) Frost LCMP LMS algorithm, direct form with diagonal loading (7.418).
- (d) MPDR-GSC-LMS with diagonal loading (7.426).
- (e) LCMP-GSC-LMS with diagonal loading (7.426).

In all cases, use the PNLMS algorithm with  $\alpha(K)$  given by (7.460) and (7.461).

The parameters to be selected for the algorithms are  $\gamma$  and  $\beta$  in (7.460) and (7.461) and the  $LNR$ . The  $SNR$  and  $INR$  must also be selected from Table 7.5.

The problem number are shown in Table 7.3. In the first ten problems do parts (a), (b), and (d). In the next five problems do parts (c) and (e). The list in Table 7.5 corresponds to Problems 7.7.1–7.7.15.

**Problem 7.7.16**

Consider a SLA10. The signal arrives from broadside with an  $SNR = 10$  dB. There are two interfering plane wave signals:

$$u_{I1} = 0.29, \quad INR_1 = 30 \text{ dB},$$

$$u_{I2} = 0.50, \quad INR_2 = 30 \text{ dB},$$

$$u_{I3} = -0.29, \quad INR_3 = 20 \text{ dB},$$

$$u_{I4} = -0.50, \quad INR_4 = 10 \text{ dB}.$$

Compare the performance of SMI, RLS, and LMS beamformers.

Assume that we use an LCMV beamformer with a single constraint

$$\mathbf{w}^H \mathbf{v}_s = 1. \quad (7.588)$$

Design an SMI beamformer with appropriate diagonal loading. Design an RLS beamformer with appropriate  $\mu$  and  $\sigma_o^2$ . Design an LMS beamformer with appropriate  $\alpha(K)$  and  $\sigma_L^2$ . Simulate your beamformers and plot the following results:

- (i)  $\overline{SINR}_o$  versus  $K$  ( $K = 10, \dots, 1000$ );
- (ii) Beam patterns for representative trials at  $K = 2N, 6N, 10N, 50N$ , and  $100N$ .

Discuss your results.

**Problem 7.7.17** (continuation, Problem 7.4.20)

Consider the same model as in Problem 7.4.20. Implement the power normalized version ((7.460) and (7.461)) of the Griffiths LMS beamformer. Try different values of  $\gamma$  and  $\beta$  and various levels of diagonal loading. Plot the average  $SINR_o$  versus  $K$  and the beampatterns for various  $K$ . Discuss your results.

**Problem 7.7.18** (continuation)

Repeat Problem 7.7.17 for an MPDR-LMS beamformer in both the direct form and GSC structure.

**Problem 7.7.19** (continuation)

Repeat Problem 7.7.18 for an MPDR-GSC-LMS beamformer using a quadratic constraint,  $T_o = 2/N$ . Compare fixed loading, variable loading, and scaled projection.

## P7.8 Detection of Signal Subspace Dimension

**Problem Note 7.8.1**

The next ten problems consider an SLA10. In each problem, plot  $P_D$ ,  $P_M$ , and  $P_{FA}$  versus  $ASNR$  for MDL, MDL-FB, AIC, AIC-FB, EDT, and EDT-FB. For the two-signal case, assume that  $u_1 = -\Delta u/2$ ,  $u_2 = \Delta u/2$ .

**Problem 7.8.1**

Consider two equal-power uncorrelated signals and four values of  $\Delta u$ ; 0.1 HPBW, 0.5 HPBW, 1.0 HPBW, and  $\Delta u = 0.3$ . Consider the low sample support case,  $K = 20$ .

**Problem 7.8.2**

Consider two uncorrelated signals with unequal-power; (i)  $ASNR_1 = 10ASNR_2$ , (ii)  $ASNR_1 = 100ASNR_2$ . Assume  $K = 100$ .

**Problem 7.8.3**

Consider two equal-power correlated signals. The values of  $\rho$  of interest are:

- (i)  $|\rho| = 0.95$ ,  $\phi_\rho = 0, \pi/4, \pi/2$ ,
- (ii)  $|\rho| = 0.99$ ,  $\phi_\rho = 0, \pi/4, \pi/2$ ,
- (iii)  $|\rho| = 1.0$ ,  $\phi_\rho = 0, \pi/4, \pi/2$ .

Assume  $K = 100$ .

**Problem 7.8.4**

Repeat Examples 7.8.5–7.8.7 for the case in which FB averaging is used.

**Problem 7.8.5**

Consider three equal-power uncorrelated signals located at  $u_1 = -0.18$ ,  $u_2 = 0$ ,  $u_3 = 0.18$ . Assume  $K = 100$ .

**Problem 7.8.6** (continuation)

Consider three uncorrelated signals located at  $u_1 = -0.18$ ,  $u_2 = 0$ ,  $u_3 = 0.18$ . Assume  $K = 100$ .

- (a) Assume  $ASNR_1 = ASNR_3 = 10ASNR_2$ .
- (b) Assume  $ASNR_1 = ASNR_3 = 100ASNR_2$ .

**Problem 7.8.7** (continuation)

Consider three equal-power correlated signals located at  $u_1 = -0.18$ ,  $u_2 = 0$ ,  $u_3 = 0.18$ . Assume  $K = 100$ . Consider various signal correlation matrices,  $\rho$ .

**Problem 7.8.8** (continuation, Example 7.8.5)

The performance of the EDT and EDT-FB algorithms depend on the location of the signals relative to  $\psi_o$  (7.514). In this problem, we consider the model in Example 7.8.5 except

$$u_1 = u_c - \Delta u/2 \quad (7.589)$$

and

$$u_2 = u_c + \Delta u/2, \quad (7.590)$$

where  $|u_c| \leq 0.1$  is a parameter and  $\Delta u = 0.1$  HPBW.

Assume  $K = 100$ . Plot  $P_D$  versus  $ASNR$  for  $u_c = 0, 0.2, 0.4, 0.6, 0.8$ , and 1.0 times  $\Delta u/2$ . Consider the EDT, BS-AIC, and BS-MDL algorithms with both forward-only (FO) and FB averaging.

**Problem 7.8.9**

Repeat Problem 7.8.8 for  $\Delta u = 0.5$  HPBW.

**Problem 7.8.10** (continuation: Problem 7.8.1)

Read the paper by Shah and Tufts [ST94]. Implement their algorithm for the model in Problem 7.8.1. Compare the results to these in Problem 7.8.1.

## P7.9 Eigenspace and DMR Beamformers

**Problem 7.9.1:** The first fifteen problems utilize the test scenarios in Table 7.6. In each problem, we use AIC-FB or MDL-FB to detect the dimension of the signal-plus-interference subspace. We then implement an MPDR or LCMP beamformer in eigenspace. We utilize SMI processing in the eigenspace beamformer. In each problem, plot

- (a)  $SINR_o$  versus  $K$ .
- (b)  $P_D$  and  $P_{FA}$  versus  $K$ .
- (c) Repeat part (a) for a DMR beamformer.

Discuss your results. The assignment by problems to test scenarios is given in Table 7.4. Problems 7.9.1–7.9.15 are described by this note.

**Problem Note 7.9.2:** The next several problems consider subspace tracking and require you to read the original references. In each problem, you must develop the subspace

tracker and then implement the eigenspace beamformer. For some of the subspace tracking algorithms, (7.520) can be used directly with an RLS or LMS adaptive implementation. In other cases, the eigenspace algorithm must be matched to the subspace tracking algorithm.

Table 7.6

| Test Scenario | Eigenspace | Beamspace | Array perturbations |
|---------------|------------|-----------|---------------------|
| 1             | 7.9.1      | 7.10.1    | 7.12.1              |
| 1m            | 7.9.2      | 7.10.2    | 7.12.2              |
| 2             | 7.9.3      | 7.10.3    | 7.12.3              |
| 2m            | 7.9.4      | 7.10.4    | 7.12.4              |
| 3             | 7.9.5      | 7.10.5    | 7.12.5              |
| 3m            | 7.9.6      | 7.10.6    | 7.12.6              |
| 4             | 7.9.7      | 7.10.7    | 7.12.7              |
| 5             | 7.9.8      | 7.10.8    | 7.12.8              |
| 6             | 7.9.9      | 7.10.9    | 7.12.9              |
| 7             | 7.9.10     | 7.10.10   | 7.12.10             |
| 1m            | 7.9.11     | 7.10.11   | 7.12.11             |
| 2m            | 7.9.12     | 7.10.12   | 7.12.12             |
| 3m            | 7.9.13     | 7.10.13   | 7.12.13             |
| 4             | 7.9.14     | 7.10.14   | 7.12.14             |
| 6             | 7.9.15     | 7.10.15   | 7.12.15             |

**Problem 7.9.16**

Read the discussion of the ROSE algorithm in DeGroat and Roberts [DR90]. Implement their algorithm and the corresponding eigenspace beamformer for the signal and interference model in Example 6.9.1. Note that the dimension of the subspace must be detected. Compare your results to the SMI results in Example 6.9.1.

**Problem 7.9.17**

Read the discussion of the OPERA algorithm in MacInnes [MacI98]. Repeat Problem 7.9.16.

**Problem 7.9.18**

Read the discussion of the FAST algorithm in Real et. al. [RTC99]. Repeat Problem 7.9.16.

**Problem 7.9.19**

Read the discussion of the LORAF algorithm in Strobach [Str96]. Repeat Problem 7.9.16.

**Problem 7.9.20**

Read the discussion of PAST and PAST-D in Yang [Yan95a], [Yan95b]. Repeat Problem 7.9.16.

**Problem 7.9.21**

A primary motivation for subspace tracking is that it enables the eigenspace beamformer to change dimension when the signal-plus-interference subspace changes.

Derive a scenario in which interferers appear and/or disappear every 100 snapshots. Test one or more of the above subspace trackers and eigenspace beamformers with the above scenario.

### P7.10 Beamspace Beamformers

#### Problem Note 7.10.1

The first sixteen problems utilize the test scenarios in Table 7.6. Use a  $5 \times 10$  DFT beamspace matrix in each problem. In each problem, implement:

- (i) SMI,
- (ii) RLS,
- (iii) LMS.

Use appropriate diagonal loading. In each problem, plot

- (a)  $SINR_o$  versus  $K$ ,
- (b) Representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

The problem assignment to test scenarios is shown in Table 7.6. In all of the beamspace problems, utilize FB averaging of the data.

#### Problem Note 7.10.2

A 32-element SLA provides more flexibility to study beamspace processing. In Table 7.7, several test scenarios are listed. Use a  $7 \times 32$  DFT beamspace matrix in each problem. In each problem, implement:

- (i) SMI,
- (ii) RLS,
- (iii) LMS.

Use appropriate diagonal loading. In each problem, plot

- (a)  $SINR_o$  versus  $K$ .
- (b) Representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

Problems 7.10.17–7.10.21 correspond to BS-MPDR implementations for the five test scenarios. Problems 7.10.22 and 7.10.23 correspond to BS-LCMP implementations for test scenarios 1m and 3m, respectively.

Table 7.7

| Test Scenario | Signal                                      | Interference                                 |  |
|---------------|---|--|--|
|               |   | Location                                     | Strength                                       |
| 1             | $u_m = 0$<br>$SNR = -10, 0, \dots, 20$ dB   | $u_1 = 3/32$<br>$u_2 = 5/32$<br>$u_3 = 7/32$ | $INR_i = 0, \dots, 40$ dB<br>$i = 1, \dots, 3$ |
| 1m            | $u_a = \pm 1/128, \pm 1/64$ , or $\pm 1/32$ | TS1  |  |
| 2             | TS1   | TS1 plus 50 dB interferer at $u_i = 27/32$   |  |
| 3             | $u_m = 0$                                   | $u_1 = 1/32$<br>$u_2 = -3/32$                |  |
| 3m            | $u_m = \pm 1/128$                           | TS3  |  |

**Problem Note 7.10.3**

The next set of problems studies the adaptive behavior of the beamformers that we studied in Problems 6.9.4–6.9.15. Problems 7.10.24–7.10.35 correspond to those problems, respectively. Problems 7.10.36 and 7.10.37 correspond to Problems 6.10.8 and 6.10.9, respectively. In each problem, implement:

- (i) SMI,
- (ii) RLS,
- (iii) LMS.

Use appropriate diagonal loading. In each problem, plot

- (a)  $SINR_o$  versus  $K$ .
- (b) Representative beam patterns for  $K = 2N, 6N, 10N$ , and  $100N$ .

**P7.11 Broadband Beamformers****Problem 7.11.1**

Consider a ULA10. The interelement spacing is  $\lambda_u/2$ , where  $\lambda_u$  is the wavelength of the highest frequency in the input signal. We use the signal and interference model in Examples 6.13.2 and 6.13.5.

- (a) Implement an FFT beamformer using uncoupled adaptive narrowband beamformers in each bin. Consider  $M = 4, 8$ , and  $16$ . Implement the adaptive narrowband beamformers using SMI, RLS, and LMS algorithms. Plot the  $SINR_o$  versus  $K$ . Discuss your results.
- (b) Repeat part (a) using a time-domain beamformer with the same degrees of freedom. Discuss your results.

**Problem 7.11.2** (continuation, Example 7.11.1)

Consider the same model as in Example 7.11.1. In each of following parts, implement a time-domain adaptive beamformer, plot  $SINR_o$  versus  $K$  and discuss your results. Simulate both SMI and LMS implementations.

- (a) Denote the number of taps by  $M$ . Simulate the beamformer for  $M = 6, 8$ , and  $10$ .
- (b) Change  $u_s$  to  $-0.10$ . Simulate the performance for various  $SNR$  and  $M$ .
- (c) Assume the desired signal arrives from  $-0.10$  with a flat spectrum with  $B_f = 0.4$ . The interfering signal arrives from  $0.29$  with a flat spectrum over the normalized frequency range  $0.6\pi < \omega < 1.0\pi$ .

**Problem 7.11.3**

Repeat Problem 7.11.2 using an FFT beamformer.

**Problem 7.11.4** (continuation, Example 6.13.5)

Consider the same model as in Example 6.13.5. Assume  $u_I = 0.5BW_{NN}$  and  $M = 2$  and  $4$ . Use a GSC configuration and implement an SMI version and an LMS version. The  $SNR = 0$  dB and the  $INR = 30$  dB. Plot the  $SINR_o$  versus  $K$  and discuss your results.

**Problem 7.11.5** (continuation, Example 6.13.6)

Consider the same model as in Example 6.13.6. Use a GSC configuration and implement an SMI version and an LMS version.

Assume the  $ASNR = 20$  dB,  $u_s = 0.05$ ,  $INR = 30$  dB,  $u_I = 0.45$ , and  $M = 4$ .

Plot the average  $SINR_o$  versus  $K$ . Discuss your results.

## P7.12 Summary

**Problem Note 7.12.1**

Problems 7.12.1–7.12.15 study the effect of array perturbations on the various algorithms. In each problem assume that the array is on the  $z$ -axis and that element locations are perturbed in the  $x$ -,  $y$ -, and  $z$ - directions using the model in (2.174) and (2.185). Let  $\sigma_\lambda = 0.02$ , 0.05, and 0.1. Repeat the problem in the same rows of Tables 7.4, 7.5, and 7.6 for the following algorithms:

- (a) SMI,
- (b) RLS,
- (c) QRD,
- (d) SD,
- (e) LMS,
- (f) Eigenspace and DMR,
- (g) Beamspace.

## Bibliography

- [AC76] S. P. Applebaum and D. J. Chapman. Adaptive arrays with main beam constraints. *IEEE Trans. Antennas Propag.*, vol.AP-24, pp. 650–662, September 1976.
- [AH89] A. H. Abdallah and Y. H. Hu. Parallel VLSI computing array implementation for signal subspace updating algorithm. *IEEE Trans. Acoust., Speech Signal Process*, vol.ASSP-37, pp. 742–748 May 1989.
- [Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, vol.AC-19, pp. 716–723, June 1974.
- [Ale86] S. T. Alexander. *Adaptive Signal Processing - Theory and Applications*. Wiley, New York, 1986.
- [And63] T. W. Anderson. Asymptotic theory for principal component analysis. *Ann. Math. Stat.*, vol.34, pp. 122–148, January 1963.
- [And73] T. W. Anderson. Asymptotically efficient estimation of covariance matrices with linear structure. *Ann. Stat.*, vol.1, pp. 135–141, January 1973.
- [And84] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, 2nd edition, 1984.
- [AO90] D.A. Abraham and N.L. Owsley. Beamforming with dominant mode rejection. *IEEE Oceans 90*, pp. 470–475, 1990.

- [App66] S.P. Applebaum. Adaptive arrays. Technical Report, Syracuse University Research Corp., August 1966.
- [App76] S.P. Applebaum. Adaptive arrays. *IEEE Trans. Antennas Propag.*, vol.AP-24, pp. 585–598, September 1976.
- [AS65] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, New York, 1965.
- [ASHP77] N. Ahmed, D. L. Soldan, D. A. Hummels, and D. D. Parikh. Sequential regression considerations of adaptive filtering. *Electron. Lett.*, p. 446, July 1977.
- [AHUS79] N. Ahmed, D. A. Hummels, M. Uhl, and D. L. Soldan. A short term sequential regression algorithm. *IEEE Trans. Acoust. Speech, Signal Process.*, vol.ASSP-27, p. 453, October 1979.
- [Bag76] A. B. Bagheroer. Space-time processes and optimal array processing. Technical Report 506, Navy Undersea Systems Center, San Diego, California, December 1976.
- [Bai73] C. A. Baird. Recursive algorithms for adaptive array antennas. Technical Report Contract F30602-72-C-0499, Rome Air Development Center, Rome, New York, September 1973.
- [Bai74] C. A. Baird. Recursive processing for adaptive arrays. In *Proc. Adaptive Antenna Systems Workshop*, Naval Research Lab., Washington, D. C., March 1974; see also NRL Report 7803.
- [Bar54] M. S. Bartlett. A note on the multiplying factors for various  $\chi^2$  approximations. *J. R. Stat. Soc.*, vol.16, pp. 296–298, 1954.
- [BG86] K. M. Buckley and L. J. Griffiths. An adaptive generalized sidelobe canceller with derivative constraints. *IEEE Trans. Antennas Propag.*, vol.AP-34, pp. 311–319, March 1986.
- [BH85] E. Brookner and J. M. Howell. Adaptive-adaptive array processing. *Proc. Phased Arrays Symp.*, pp. 133–134, 1985.
- [BLW82] J. P. Burg, D. G. Luenberger, and D. L. Wenger. Estimation of structured covariance matrices. *Proc. IEEE*, vol.70, pp. 963–974 September 1982.
- [BN78] J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. *Numer. Math.*, vol.31, pp. 111–129, 1978.
- [BNS78] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numer. Math.*, vol.31, pp. 31–48, 1978.
- [BOBH69] A. H. Booker, C. Y. Ong, J. P. Burg, and G. D. Hair. Multiple-constraint adaptive filtering. Technical Report, Texas Instruments, Science Services Division, Dallas, Texas, April 1969.
- [Bor80] D.M. Boroson. Sample size considerations for adaptive arrays. *IEEE Trans. Aerospace and Electron Syst.*, vol.AES-16, pp. 446–451, July 1980.
- [Bri81] D. R. Brillinger, Editor. *Time Series: Data Analysis and Theory* (expanded ed.). Holden-Day, San Francisco, California, 1981.
- [BS92] C. H. Bischof and G. M. Shroff. On updating signal subspaces. *IEEE Trans. Signal Process.*, vol.SP-40, pp. 96–105, January 1992.

- [Buc91] R. S. Bucy. Geometry and multiple direction estimation. *Inform. Sci.*, vol.57-58, pp. 145–58, 1991.
- [Cad88] J. A. Cadzow. Signal enhancement – A composite property mapping algorithm. *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. 36, pp. 49–62, January 1988.
- [Car88] B. D. Carlson. Covariance matrix estimation errors and diagonal loading in adaptive arrays. *IEEE Trans. Aerospace and Elec. Syst.*, vol.AES-24, pp. 397–401, July 1988.
- [CB76] A. Cantoni and P. Butler. Eigenvalues and eigenvectors of symmetric centrosymmetric matrices. *Linear Algebra and its Applications*, vol.13, pp. 275–288, 1976.
- [CF56] A. G. Carlton and J. W. Follin. Recent developments of fixed and adaptive filtering. Technical Report 21, AGAR Dograph, 1956.
- [CG70] J. Capon and N. R. Goodman. Probability distributions for estimators of the frequency-wavenumber spectrum. *Proc. IEEE*, vol.58, pp. 1785–1786, October 1970.
- [CG90] P. Comon and G. Golub. Tracking a few extreme singular values and vectors in signal processing. *Proc. IEEE*, vol.78, pp. 1327–1343, August 1990.
- [Cha87] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra, Appl.*, vol.88/89, pp. 67–82, 1987.
- [Cha94] B. Champagne. Adaptive eigendecomposition of data covariance matrices based on first-order perturbations. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 2758–2770, October 1994.
- [Com88] R. T. Compton, Jr. *Adaptive Antennas (Concepts and Performance)*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [CP97] H. Cox and R. Pitre. Robust DMR and multi-rate adaptive beamforming. *Proc. 31st Asilomar Conf. on Signals, Systems and Computer*, Pacific Grove, California, November 1997.
- [CPL98] H. Cox, R. Pitre, and H. Lai. Robust adaptive matched field processing. *Proc. 32nd Asilomar Conf. on Signals, Systems and Computer*, Pacific Grove, California, November 1998.
- [CY92] L. Chang and C. Yeh. Performance of DMI and eigenspace-based beamformers. *IEEE Trans. Antennas Propag.*, vol.AP-40, pp. 1336–1347, November 1992.
- [CZO87] H. Cox, R. M. Zeskind, and M. M. Owen. Robust adaptive beamforming. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-35, pp. 1365–1376, October 1987.
- [DAD94] E. M. Dowling, L. P. Ammann, and R. D. DeGroat. A TQR-iteration based adaptive SVD for real-time angle and frequency tracking. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 914–926, April 1994.
- [DCA95] P. S. R. Diniz, M. L. R. de Campos, and A. Antoniou. Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor. *IEEE Trans. Signal Process.*, vol.SP-43, pp. 617–627, March 1995.

- [Deg87] S. Degerine. Maximum likelihood estimation of autocovariance matrices from replicated short time series. *J. Time Ser. Anal.*, Vol. 8, pp. 135–146, 1987.
- [DeG92] R. D. DeGroat. Noniterative subspace tracking. *IEEE Trans. Signal Process.*, vol.SP-40, pp. 571–577, March 1992.
- [DeGR87] R. D. DeGroat and R. A. Roberts. A family of rank-one eigenstructure updating methods. *Proc. 21st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, California, November 1987.
- [Dep88] E. F. Deprettere, Editor. *SVD and Signal Processing: Algorithms, Applications and Architectures*. North-Holland, Amsterdam 1988.
- [DMS89] A. Dembo, C. L. Mallows, and L. A. Shepp. Embedding nonnegative-definite Toeplitz matrices in nonnegative-definite circulant matrices, with application to covariance estimation. *IEEE Trans. Inf. Theory*, vol.IT-35, pp. 1206–1212, November 1989.
- [DR90] R. D. DeGroat and R. A. Roberts. Efficient, numerically stabilized rank-one eigenstructure updating. *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol.ASSP-38, pp. 301–316, February 1990.
- [EC85] M. H. Er and A. Cantoni. An alternative formulation for an optimum beamformer with robustness capability. *Proc. IEE*, vol.132, pp. 447–460, October 1985.
- [EC86] M. H. Er and A. Cantoni. An unconstrained partitioned realization for derivative constrained broad-band antenna array processors. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-34, pp. 1376–1379, December 1986.
- [EJS82] J.E. Evans, J.R. Johnson, and D.F. Sun. Application of advanced signal processing techniques to angle of arrival estimation in ATC navigation and surveillance systems. Technical report, M.I.T. Lincoln Laboratory, Lexington, Massachusetts, June 1982.
- [EY36] G. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, vol.1, pp. 211–218, 1936.
- [Far83] D.R. Farrier. Gain of an array of sensors subjected to processor perturbation. *Proc. Inst. Elec. Eng.*, vol.72, pp. 251–254, June 1983.
- [FG94] D. D. Feldman and L. J. Griffiths. A projection approach for robust adaptive beamforming. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 867–876, April 1994.
- [Fis12] R. A. Fisher. On an absolute criterion for fitting frequency curves. *Mess. of Math.*, vol.41, pp. 155–165, 1912.
- [FM88] D. R. Fuhrmann and M. I. Miller. On the existence of positive-definite maximum-likelihood estimates of structured covariance matrices. *IEEE Trans. Inf. Theory*, vol.IT-34, pp. 722–729, July 1988.
- [FM97] D. R. Fuhrmann and M. I. Miller. Correction to “On the existence of positive-definite maximum-likelihood estimates of structured covariance matrices.” *IEEE Trans. Inf. Theory*, vol.IT-43, pp. 1094–1096, May 1997.
- [Fro72] O. L. Frost III. An algorithm for linearly constrained adaptive array processing. *Proc. IEEE*, vol.60, pp. 926–935, August 1972.

- [FSM96] D. R. Fuhrmann, A. Srivastava, and H. Moon. Subspace tracking via rigid body dynamics. *Proc. 8th Stat. Signal Array Processing Workshop*, Corfu, Greece, pp. 578–581, June 1996.
- [FTM88] D. R. Fuhrmann, M. J. Turmon, and M. I. Miller. Efficient implementation of the EM algorithm for Toeplitz covariance estimation. *Proc. Annu. Conf. Information Science and Systems*, Princeton, New Jersey, March 1988.
- [Fuh88] D. R. Fuhrmann. Progress in structured covariance estimation. *Proc. 4th ASSP Workshop on Spectrum Estimation and Modeling*, pp. 158–161, Minneapolis, Minnesota, August 1988.
- [Fuh91] D. R. Fuhrmann. Application of Toeplitz covariance estimation to adaptive beamforming and detection. *IEEE Trans. on Signal Process.*, vol. 39, pp. 2194–2198, October 1991.
- [Fuh98] D. R. Fuhrmann. A geometric approach to subspace tracking. *Proc. 31st Asilomar Conf. on Signals, Systems, and Computers*, vol.1, pp. 783–787, Pacific Grove, California, November 1998.
- [Gab85] W. F. Gabriel. Using spectral estimation techniques in adaptive array systems. *Proc. of the Phased Arrays 1985 Symp.*, pp. 109–113, Rome, New York, 1985.
- [GAS87] D. Gray, B. Anderson, and P. Sim. Estimation of structured covariances with application to array beamforming. *Circuits, Syst., Signal Process.*, vol.6, pp. 421–447, June 1987.
- [GBT99] G. O. Glentis, K. Berberidis, and S. Theodoridis. Efficient least squares adaptive algorithms for FIR transversal filtering. *IEEE Signal Process. Mag.*, vol.16, pp. 13–41, January 1999.
- [Gen73] W. M. Gentleman. Least-squares computation by Givens transformations without square-roots. *J. Inst. Math. Appl.*, vol.12, pp. 329–369, 1973.
- [GJ82] L. J. Griffiths and C. W. Jim. An alternative approach to linearly constrained adaptive beamforming. *IEEE Trans. Antennas Propag.*, vol.AP-30, pp. 27–34, January 1982.
- [GMW90] M. W. Ganz, R. L. Moses, and S. L. Wilson. Convergence of the SMI and the diagonally loaded SMI algorithms with weak interference. *IEEE Trans. Antennas Propag.*, vol.AP-38, pp. 394–399, March 1990.
- [God74] D. Godard. Channel equalization using a Kalman filter for fast data transmission. *IBM J. Res. Dev.*, vol.18, pp. 267–273, May 1974.
- [God85] L.C. Godara. The effect of phase-shifter error and the performance of an antenna array beamformer. *IEEE J. Oceanogr. Eng.*, vol.OE-10, pp. 278–284, July 1985.
- [God86] L. C. Godara. Error analysis of the optimal antenna array processors. *IEEE Trans. Aerosp. Electron. Syst.*, vol.AES-22, pp. 395–409, July 1986.
- [Gol73] G. H. Golub. Some modified matrix eigenvalue problems. *SIAM Rev.*, vol.15, pp. 318–334, 1973.
- [Goo63] N. R. Goodman. Statistical analysis based on a certain complex Gaussian distribution. *Ann. Math. Stat.*, vol.34, pp. 152–180, 1963.

- [GR80] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, New York, 1980.
- [Gra72] D. Graupe. *Identification of Systems*, chapter 6, Van Nostrand Reinhold, New York, 1972.
- [Gri68] L. J. Griffiths. *Signal extraction using real-time adaptation of a linear multi-channel filter*. PhD thesis, Stanford University, Stanford, California, January 1968.
- [Gri69] L.J. Griffiths. A simple adaptive algorithm for real-time processing in antenna arrays. *Proc. IEEE*, vol.57, pp. 1696–1704, October 1969.
- [Gri77] L.J. Griffiths. A continuously-adaptive filter implemented as a lattice structure. *Proc. ICASSP*, Hartford, Connecticut, May 1977.
- [Gri78] L.J. Griffiths. An adaptive lattice structure for noise-cancelling applications. *Proc. ICASSP*, Tulsa, Oklahoma, pp. 87-90, 1978.
- [GS84] G. Goodwin and K. Sin. *Adaptive Filtering, Prediction and Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1984.
- [GSB96] A. B.Gershman, G. V. Serebryakov, and J. F. Bohme. Constrained Hung-turner adaptive beamforming algorithm with additional robustness to wideband and moving jammers. *IEEE Trans. Antennas Propag.*, vol.AP-44, pp. 361–367, 1996.
- [Gue99] J. R. Guerci. Theory and application of covariance matrix tapers for robust adaptive beamforming. *IEEE Trans. Signal Process.*, vol.SP-47, pp. 977–985, April 1999.
- [Gue00] J. R. Guerci. Reply to “Comments on ‘Theory and application of covariance matrix tapers for robust adaptive beamforming.’” *IEEE Trans. Signal Process.*, vol.SP-48, p. 1800, June 2000.
- [Gup65] R. P. Gupta. Asymptotic theory for principal component analysis in the complex case. *J. Indian Stat. Assoc.*, vol.3, pp. 97–106, 1965.
- [Ham74] S. Hammarling. A note on modifications to the Givens plane rotation. *J. Inst. Math. Appl.*, vol.3, pp. 215–218, 1974.
- [Hay85] S. Haykin. *Array Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
- [Hay91] S. Haykin. *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, New Jersey, 2nd edition, 1991.
- [Hay96] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 1996.
- [HM84] M. L. Honig and D. G. Messerschmitt. *Adaptive Filters - Structures, Algorithms and Applications*. Kluwer, Dordrecht, The Netherlands, 1984.
- [How76] P. W. Howells. Explorations in fixed and adaptive resolution at GE and SURC. *IEEE Trans. Antennas Propag.*, vol.AP-24, pp. 575–584, September 1976.
- [HS92] S. Haykin and A. Steinhardt, Editors. *Adaptive Radar Detection and Estimation*. Wiley, New York, 1992.

- [Hud79] J. E. Hudson. A Kalman type algorithm for adaptive radar arrays and modelling of non-stationary weights. In *IEE Conf. Publ.*, p. 180, 1979; also, in “Case Studies in Advanced Signal Processing.”
- [Hud81] J.E. Hudson. *Adaptive Array Principles*. Peter Peregrinus, New York and London, 1981.
- [HY91] K.-C. Huarng and C.-C. Yeh. Adaptive beamforming with conjugate symmetric weights. *IEEE Trans. Antennas Propag.*, vol.AP-39, pp. 926–932, July 1991.
- [IS71] F. Itakura and S. Saito. Digital filtering techniques for speech analysis and synthesis. *Proc. 7th Int. Conf. on Acoustics*, vol.3, paper 25C-1, pp. 261–264, 1971.
- [Jim77] C. W. Jim. A comparison of two LMS constrained optimal array structures. *Proc. IEEE*, vol.65 pp. 977–978, December 1977.
- [JS99] M. Jansson and P. Stoica. Forward-only and forward-backward sample covariances—A comparative study. *Signal Process.*, vol.77, pp. 235–245, 1999.
- [Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *J. Basic Eng.*, vol.82D, pp. 35–45, March 1960.
- [Kar86] I. Karasalo. Estimating the covariance matrix by signal subspace averaging. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-34, pp. 8–12, January 1986.
- [Kay88] S. M. Kay. *Modern Spectral Estimation*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [KB61] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng.*, vol.83D, pp. 95–108, 1961.
- [Kel86] E. J. Kelly. An adaptive detection algorithm. *IEEE Trans. Aerospace Electron. Syst.*, vol.AES-22, pp. 115–127, March 1986.
- [Kel87a] E. J. Kelly. Adaptive detection in non-stationary interference, Part III. Technical Report 761, M.I.T. Lincoln Laboratory, Lexington, Massachusetts, August 1987.
- [Kel87b] E. J. Kelly. Performance of an adaptive detection algorithm; rejection of unwanted signals. *IEEE Trans. Aerospace Electron. Syst.*, vol.AES-25, pp. 122–133, March 1987.
- [KF89] E. J. Kelly and K. M. Forsythe. Adaptive detection and parameter estimation for multidimensional signal models. Technical Report, M.I.T. Lincoln Laboratory, Lexington, Massachusetts, April 1989.
- [KL80] V. C. Klema and A. J. Laub. The singular value decomposition: Its computation and some applications. *IEEE Trans. Autom. Control*, vol.AC-25, pp. 164–176, April 1980.
- [Kle80] L. I. Kleinberg. Array gain for signals and noise having amplitude and phase fluctuations. *J. Acoust. Soc. Am.*, vol.67, pp. 572–576, February 1980.
- [Kol41a] A. N. Kolmogorov. Interpolation and extrapolation of stationary random sequences. Technical Report RM-3090-PR, RAND Corp., Santa Monica, California, 1941. Translated by W. Doyle and J. Selin in 1962.

- [Kol41b] A. N. Kolmogorov. Interpolation and extrapolation von stationären zufälligen folgen. *Bull. Acad. Sci. USSR Ser. Math.*, vol.5, pp. 3–14, 1941.
- [KT93] N. Kalouptsidis and S. Theodoridis. *Adaptive System Identification and Signal Processing Algorithms*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
- [Kus84] H. J. Kushner. Approximation and weak convergence methods for random processes with applications to stochastic system theory. MIT Press, Cambridge, Massachusetts, 1984.
- [KW91] M. Kaveh and H. Wang. Threshold properties of narrow-band signal-subspace array processing methods. In S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing*, vol.II, pp. 173–220, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [Law56] D. N. Lawley. Tests of significance of the latent roots of the covariance and correlation matrices. *Biometrika*, vol.43, pp. 128–136, 1956.
- [LDD94] D. A. Linebarger, R. D. DeGroat, and E. M. Dowling. Efficient direction-finding methods employing forward/backward averaging. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 2136–2145, August 1994.
- [Lee66] R. C. K. Lee. *Optimal Estimation, Identification, and Control*, MIT Press, Cambridge, Massachusetts, 1966.
- [Lee92] H. B. Lee. Eigenvalues and eigenvectors of covariance matrices for signals closely spaced in frequency. *IEEE Trans. Signal Process.*, vol.SP-40, pp. 2518–2535, October 1994.
- [LH74] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
- [Lig73] W. S. Liggett. Passive sonar: Fitting models to multiple time series. J.W.R. Griffiths and P.L. Stocklin, Editors, *Signal Processing, Proc. NATO ASI Signal Processing*, pp. 327–345, Academic Press, New York, 1973.
- [LL94] H. Lee and F. Li. An eigenvector technique for detecting the number of emitters in a cluster. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 2380–2388, September 1994.
- [LSL98] H. Li, P. Stoica, and J. Li. Capon estimation of covariance sequences. *Circuits, Syst., Signal Process.*, vol.17, pp. 29–49, January 1999.
- [LSL99] H. Li, P. Stoica, and J. Li. Computationally efficient maximum likelihood estimation of structured covariance matrices. *IEEE Trans. Signal Process.*, vol.47, pp. 1314–1323, May 1999.
- [Mac95] O. Macchi. *Adaptive Processing: The LMS Approach with Applications in Transmission*. Wiley, New York, 1995.
- [MacI98] C. S. MacInnes. Fast, accurate subspace tracking using operator restriction analysis. *Proc. ICASSP*, pp. 1357–1360, Seattle, Washington, 1998.
- [Mar87] S. L. Marple, Jr. *Digital Spectral Analysis With Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [Mau40] J. W. Mauchley. Significance test for sphericity of a normal n-variate distribution. *Ann. Math. Stat.*, vol.11, pp. 204–209, 1940.

- [Mer65] H. Mermoz. *Adaptive filtering and optimal utilization of an antenna*. Ph.D. thesis, U.S. Navy Bureau of Ships (translation 903 of Ph.D. thesis, Institut Polytechnique, Grenoble, France), Washington D.C., October 1965.
- [MFOS91] M. I. Miller, D. R. Fuhrmann, J. A. O'Sullivan, and D. L. Snyder. Maximum-likelihood methods for Toeplitz covariance estimation and radar imaging. In S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing*, vol. II, pp. 145–172. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [MM80] R. A. Monzingo and T. W. Miller. *Introduction to Adaptive Arrays*. Wiley, New York, 1980.
- [Mos70] J. L. Moschner. Adaptive filtering with clipped input data. Technical Report, Doc. SEL-70-053, Stanford Electronics Laboratories, Stanford, California, June 1970.
- [MP81] R.A. Mucci and R.G. Pridham. Impact of beam steering errors on shifted sideband and phase shift beamforming techniques. *J. Acoust. Soc. Am.*, vol.69, pp. 1360–1368, May 1981.
- [MS87] H. Messer and P. M. Schultheiss. Estimation of source location and spectral parameters with arrays subject to coherent interference. *11th Colloq. Gretsi*, Nice, France, June 1987.
- [MS87b] M. I. Miller and D. L. Snyder. The role of likelihood and entropy in incomplete-data problems: applications to estimating point-process intensities and Toeplitz constrained covariances. *Proc. IEEE*, vol.75, July 1987.
- [MS89] J. G. McWhirter and T. J. Shepherd. Systolic array processor for MVDR beamforming. *IEE Proc. (London)*, part F, vol.136, pp. 75–80, 1989.
- [Mui82] R. J. Muirhead. *Aspects of Multivariate Statistical Theory*. Wiley, New York, 1982.
- [MV78] J. Makhoul and R. Viswanathan. Adaptive lattice methods for linear prediction. *Proc. ICASSP*, vol.1, pp. 87–90, Tulsa, Oklahoma, April 1978.
- [MV96] C. S. MacInnes and R. J. Vaccaro. Tracking directions-of-arrival with invariant subspace updating. *Proc. ICASSP*, vol.4, pp. 2896–2899, Atlanta, Georgia, 1996.
- [Nit76] R. Nitzberg. Effect of errors in adaptive weights. *IEEE Trans. Aerosp. Electron. Syst.*, vol.AES-12, pp. 369–374, May 1976.
- [Nit80] R. Nitzberg. Application of maximum likelihood estimation of persymmetric covariance matrices to adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.*, vol.AES-16, pp. 124–127, January 1980.
- [Nut76] A. Nuttall. Spectral analysis of a univariate process with bad data points, via maximum entropy and linear predictive techniques. Technical Report 5303, Naval Undersea Systems Center, San Diego, California, March 1976.
- [Orf88] S. J. Orfanidis. *Optimum Signal Processing*. McGraw-Hill, New York, 2nd edition, 1988.
- [Ows73] N. L. Owsley, Editor, A recent trend in adaptive spatial processing for sensor arrays: Constrained adaptation. *Proc. of NATO Advanced Study Institute on Signal Processing*, editor, *Signal Processing*, pp. 591–604, Academic Press, London and New York, 1973.

- [Ows78] N. L. Owsley. Adaptive data orthogonalization. *Proc. ICASSP*, vol.1, pp. 109–112, Tampa, Florida, 1978.
- [PA78] D. Parikh and N. Ahmed. A sequential regression algorithm for recursive filters. *Electron. Lett.*, p. 266, April, 1978.
- [PK89] S. U. Pillai and B. H. Kwon. Forward/backward spatial smoothing techniques for coherent signal identification. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-37, pp. 8–15, January 1989.
- [PRLN92] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias. *Advanced Digital Signal Processing*. Macmillan Publishing Company, New York, 1992.
- [Qua82] A. H. Quazi. Array beam response in the presence of amplitude and phase fluctuations. *J. Acoust. Soc. Am.*, vol.72, pp. 171–180, July 1982.
- [Rao46] C. R. Rao. *Linear Statistical Inference and Its Applications*. Wiley, New York, 1946.
- [Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, vol.14, pp. 465–471, 1978.
- [RMB74] I.S. Reed, J.D. Mallett, and L.E. Brennan. Rapid convergence rate in adaptive arrays. *IEEE Trans. Aerospace and Electron. Syst.*, vol.AES-10, pp. 853–863, November 1974.
- [Ros60] J. B. Rosen. The gradient projection method for nonlinear programming, pt. 1: Linear constraints. *J. Soc. Ind. Appl. Math.*, vol.8, p. 181, 1960.
- [RTC99] E. C. Real, D. W. Tufts, and J. W. Cooley. Two algorithms for fast approximate subspace tracking. *IEEE Trans. Signal Process.*, vol.47, pp. 1936–1945, July 1999.
- [Rut69] H. Rutishauser. Simultaneous iteration method for symmetric matrices. *Numer. Math.*, vol.13, pp. 4–13, 1969.
- [SC97] P. Stoica and M. Cedervall. Detection tests for array processing in unknown correlated noise fields. *IEEE Trans. Signal Process.*, vol.45, pp. 2351–2362, September 1997.
- [Sch78] G. Schwartz. Estimating the dimension of a model. *Ann. Stat.*, vol.6, pp. 461–464, 1978.
- [Sch86] R. Schreiber. Implementation of adaptive array algorithms. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-34, pp. 1038–1045, October 1986.
- [Sch91] L. L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, Reading, Massachusetts, 1991.
- [Sho66] S. W. W. Shor. Adaptive technique to discriminate against coherent noise in a narrow-band system. *J. Acoust. Soc. Am.*, vol.39, pp. 74–78, January 1966.
- [Sib87] L. H. Sibul, editor. *Adaptive Signal Processing*. IEEE Press, New York, 1987.
- [Sim80] D. N. Simkins. *Multichannel Angle-of-Arrival Estimation*. Ph.D. thesis, Stanford University, Stanford, California, 1980.
- [SK85] T. Shan and T. Kailath. Adaptive beamforming for coherent signals and interference. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-33, pp. 527–536, 1985.

- [SK94] A. H. Sayed and T. Kailath. Constraints for maximally flat optimum broadband antenna arrays. *IEEE Signal Process. Mag.*, vol.11, pp. 18–60, 1994.
- [SM97] P. Stoica and R. L. Moses. *Introduction to Spectral Analysis*. Prentice-Hall, Upper Saddle River, New Jersey, 1997.
- [Smi93] S. T. Smith. *Geometric Optimization Methods for Adaptive Filtering*. Ph.D. dissertation, Harvard University, Cambridge, Massachusetts, May 1993.
- [Sor70] H. W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, vol.7, pp. 63–68, July 1970.
- [Sri00] A. Srivastava. A Bayesian approach to geometric subspace estimation. *IEEE Trans. Signal Process.*, vol.SP-48, pp. 1390–1400, May 2000.
- [SS89] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
- [Ste75] G. W. Stewart. Methods of simultaneous iteration for calculating eigenvectors of matrices. *Topics in Numerical Analysis II*, J. H. Miller, Editor, pp.169–185, Academic Press, New York, 1975.
- [Ste76] G. W. Stewart. Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. *Numer. Math.*, vol.25, pp. 123–136, 1976.
- [Ste92] G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Trans. Signal Process.*, vol.SP-40, pp. 1535–1541, June 1992.
- [ST94] A. A. Shah and D. W. Tufts. Determination of the dimension of a signal subspace from short data records. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 2531–2535, September 1994.
- [Str96] P. Strobach. Low-rank adaptive filters. *IEEE Trans. Signal Process.*, vol.SP-44, pp. 2932–2947, December 1996.
- [Swc58] P. Swerling. A proposed stagewise differential correction procedure for satellite tracking and prediction. Technical Report P-129, RAND Corp., Santa Monica, California, January 1958.
- [Swe59] P. Swerling. A proposed stagewise differential correction procedure for satellite tracking and prediction. *J. Astronaut. Sci.*, vol.6, 1959.
- [TBV01] Z. Tian, K. L. Bell, and H. L. Van Trees. A recursive least squares implementation for LCMP beamforming under quadratic constraint. *IEEE Trans. Signal Process.*, vol.49, pp. 1138–1145, June 2001.
- [Tia99] Z. Tian. On subspace tracking. Internal Report, Array Processing Lab, George Mason University, Fairfax, Virginia, 1999.
- [TJL87] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore. *Theory and Design of Adaptive Filters*. Wiley, New York, 1987.
- [TK86] K. Takao and N. Kikuma. Tamed adaptive antenna array. *IEEE Trans. Antennas Propag.*, vol.AP-34, pp. 388–394, March 1986.
- [TK87] K. Takao and N. Kikuma. Adaptive array utilizing an adaptive spatial averaging technique for multipath environment. *IEEE Trans. Antennas Propag.*, vol.AP-35, pp. 1388–1396, December 1987.
- [TM86] D. W. Tufts and C. D. Melissinos. Simple, effective computation of principal eigenvectors and their eigenvalues and application to high-resolution estimation of frequencies. *IEEE Trans. Acoust., Speech Signal Process.*, vol.ASSP-34, pp. 1046–1053, October 1986.

- [TM94] M. J. Turmon and M. I. Miller. Maximum-likelihood estimation of complex sinusoids and Toeplitz covariances. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 1074–1086, May 1994.
- [TRC97] D. W. Tufts, E. C. Real, and J. W. Cooley. Fast approximate subspace tracking (FAST). *Proc. ICASSP*, vol.1, pp. 547–560, Munich, Germany, 1997.
- [Tuf98] D. W. Tufts. A perspective on the history of underwater acoustic signal processing. *IEEE Signal Process. Mag.*, pp. 23–27, July 1998.
- [TW96] K. W. Tam and Y. Wu. On the rate of convergence of some consistent estimates of the number of signals. *8th IEEE Signal Processing Workshop*, pp. 44–47, Corfu, Greece, 1996.
- [Ube99] J. Uber. *Detection of Signals*. Ph.D. thesis proposal, George Mason University, Fairfax, Virginia, December 2001.
- [VT68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. Wiley, New York, 1968.
- [VT71] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part III*. Wiley, New York, 1971.
- [VT01a] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. Wiley Interscience, New York, 2001.
- [VT01b] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part III*. Wiley Interscience, New York, 2001.
- [VV91a] B. D. Van Veen. Adaptive convergence of linearly constrained beamformers based on the sample covariance matrix. *IEEE Trans. Signal Process.*, vol.SP-39, pp. 1470–1473, June 1991.
- [VV91b] B. D. Van Veen. Adaptive convergence of linearly constrained beamformers based on the sample covariance matrix. Technical Report, Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin, 1991.
- [VW88] B. Van Veen and B. Williams. Structured covariance matrices and dimensional reduction in array processing. *Proc. 4th ASSP Workshop on Spectrum Estimation and Modeling*, pp. 168–171, Minneapolis, Minnesota, August 1988.
- [WA96] M. Wax and Y. Anu. Performance analysis of the minimum variance beamformer in the presence of steering vector errors. *IEEE Trans. Signal Process.*, vol.SP-44, pp. 938–947, April 1996.
- [WH60] B. Widrow and M. E. Hoff, Jr. Adaptive switching circuits. *IRE WESCON Conv. Rec.*, vol.4, pp. 96–104, 1960.
- [WH88] D. M. Wilkes and M. H. Hayes. Iterated Toeplitz approximation of covariance matrices. *Proc. ICASSP*, pp. 1663–1666, New York, April 1988.
- [Wie49] N. Wiener. *The Extrapolation , Interpolation and Smoothing of Stationary Time Series*. Wiley, New York, 1949.
- [Wil62] S. S. Wilks. *Mathematical Statistics*. Wiley, New York, 1962.
- [Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, New York, 1965.

- [Wil86] D. B. Williams. *Eigenvalue analysis for source detection with narrowband passive arrays*. Master's thesis, Rice University, Houston, Texas, September 1986.
- [Wil92] D. B. Williams. Comparison of AIC and MDL to the minimum probability of error criterion. *Proc. Sixth SSAP Workshop in Statistical Signal & Array Processing*, Victoria, Canada, pp. 114–117, October 1992.
- [Wil94] D. B. Williams. Counting the degrees of freedom when using AIC and MDL to detect signals. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 3282–3284, November 1994.
- [WJ90] D. B. Williams and D. H. Johnson. Using the sphericity test for source detection with narrow-band passive arrays. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-38, pp. 2008–2014, November 1990.
- [WJ93] D. B. Williams and D. H. Johnson. Robust estimation of structured covariance matrices. *IEEE Trans. Signal Process.*, vol.41, pp. 2891–2906, September 1993.
- [WK85] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-33, pp. 387–392, April 1985.
- [WMB75] B. Widrow, J. McCool, and M. Ball. The complex LMS algorithm. *Proc. IEEE*, vol.63, pp. 719–720, April 1975.
- [WMGG67] B. Widrow, P.E. Mantey, L.J. Griffiths, and B.B. Goode. Adaptive antenna systems. *Proc. IEEE*, vol.55, pp. 2143–2159, December 1967.
- [WS74] L. P. Winkler and M. Schwartz. Constrained array optimization by penalty function techniques. *J. Acoust. Soc. Am.*, vol.55, pp. 1042–1048, May 1974.
- [WS85] B. Widrow and S.D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
- [WT01] Y. Wu and K. W. Tam. On determination of the number of signals. *Proceedings of the International Association of Science and Technology for Development International Conference, Signal and Image Processing*, pp. 113–117, Honolulu, Hawaii, August 2001.
- [WZRY90] K. M. Wong, Q. Zhang, J .P. Reilly, and P. C. Yip. On information theoretic criteria for determining the number of signals in high resolution array processing. *IEEE Trans. Acoustics, Speech, Signal Process.*, vol.38, pp. 1959–1971, November 1990.
- [XPK92] W. Xu, J. Pierre, and M. Kaveh. Practical detection with calibrated arrays. *Proc. Sixth IEEE-SP Workshop Statistical Signal Array Processing*, Victoria, Canada, October 1992.
- [XRK94] G. Xu, R. H. Roy, and T. Kailath. Detection of number of sources via exploitation of centro-symmetry property. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 102–112, January 1994.
- [Yan95a] B. Yang. Projection approximation subspace tracking. *IEEE Trans. Signal Process.*, vol.SP-43, pp. 95–107, January 1995.
- [Yan95b] B. Yang. An extension of the PASTd algorithm to both rank and subspace tracking. *IEEE Signal Process. Lett.*, vol.2, pp. 179–182, September 1995.

- [YC86] D. H. Youn and B. Chang. Multichannel lattice filter for an adaptive array processor with linear constraints. *Proc. ICASSP*, Tokyo, Japan, pp. 1829–1832, 1986.
- [YK88] J. Yang and M. Kaveh. Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-36, pp. 241–251, February 1988.
- [YL96] S.-J. Yu and J.-H. Lee. The statistical performance of eigenspace-based adaptive array beamformers. *IEEE Trans. Antennas Propag.*, vol.AP-44, pp. 665–671, May 1996.
- [YU94] W. S. Youn and C. K. Un. Robust adaptive beamforming based on the eigenstructure method. *IEEE Trans. Signal Process.*, vol.SP-42, pp. 1543–1547, June 1994.
- [Yue91] S. M. Yuen. Exact least squares adaptive beamforming using an orthogonalization network. *IEEE Trans. Aerospace and Electron. Syst.*, vol.AES-27, pp. 311–330, March 1991.
- [Zat99] M. Zatman. The relationship between covariance matrix tapers and multiple constraints, Technical Report, M.I.T. Lincoln Laboratory, Lexington, Massachusetts, pp. 1–6, February 1999.
- [Zat00] M. Zatman. Comments on “Theory and application of covariance matrix tapers for robust adaptive beamforming.” *IEEE Trans. Signal Process.*, vol.SP-48, pp. 1796–1800, June 2000.
- [ZKB86] L. C. Zhao, P. R. Krishnaiah, and Z. D. Bai. On detection of the number of signals in presence of white noise. *J. Multivariate Anal.*, vol.20, pp. 1–20, 1986.
- [ZKB87] L. C. Zhao, P. R. Krishnaiah, and Z. D. Bai. Remarks on certain criteria for detection of number of signals. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.ASSP-35, pp. 129–132, February 1987.
- [ZKS93] M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein. Beamspace root-MUSIC. *IEEE Trans. Signal Process.*, vol.SP-41, pp. 344–364, January 1993.

Copyright of Optimum Array Processing is the property of John Wiley & Sons, Inc. 2002 and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.