

HIGH-RESOLUTION NONPARAMETRIC SPECTRAL ANALYSIS: THEORY AND APPLICATIONS

Erik G. Larsson and Jian Li, *University of Florida*
Petre Stoica, *Uppsala University*

Spectral estimation can be defined as the art of recovering the frequency content in a measured signal, and is a highly relevant problem in practice. In particular, many engineering problems, ranging from synthetic aperture radar (SAR) imaging to the analysis of time-series observed in seismology or astronomy, can be cast as a spectral analysis problem. A *nonparametric* spectral estimator is a method that, unlike parametric methods, attempts to compute the spectral content of a signal without using any a priori information or making any explicit assumption about it. While many smoothed versions of the discrete Fourier transform can be interpreted as nonparametric methods, a body of recent work has suggested more advanced such methods that are based on adaptive filterbanks. This chapter provides a review of some of the existing work in the area of nonparametric spectral estimation, including fast implementations of the most successful estimators as well as various extensions to spectral analysis of incomplete data. Numerical examples are provided to exemplify the various algorithms.

4.1 Introduction

Spectral analysis is an important problem that occurs in many applications. For instance, the task of detecting the periodicity in a time-series can be cast as a spectral analysis problem [1], and imaging using synthetic aperture radar (SAR) or other tomographic techniques can essentially be interpreted as the problem of estimating the amplitudes and frequencies of superimposed sinusoids buried in noise (see, e.g., [2], [3] and the references therein). Spectral estimation is also relevant in other problems related to radar [4]. Consequently, an abundance of literature on the topic exists; see, for example, the textbooks [5], [6], [7] and their bibliographies.

The classical approaches to spectral analysis (of uniformly sampled data) include the discrete Fourier transform (DFT) as well as its variants, which are typically

based on smoothing the DFT spectral estimate or windowing the data [5], [6], [7]. While the spectral resolution of the DFT based algorithms is usually rather poor, these methods do not make any a priori assumptions on the spectral shape and consequently they are very robust. On the other hand, there exist several parametric high-resolution methods that exhibit an excellent capability to resolve spectral peaks. This class of methods includes MUSIC [8], and other so-called subspace-based methods [5], [9]. In contrast to the DFT-based methods, these parametric techniques rely heavily on the assumption that the observed data consist of a known number of sinusoids in white noise, and unfortunately they often fail in applications where this assumption is not perfectly satisfied.

An emerging class of methods that can overcome the resolution limit of the DFT, and to some extent trade spectral resolution against robustness and statistical stability is the group of so-called *nonparametric adaptive filter-bank methods*. This class of estimators contains the classical Capon algorithm [10], and the more recent amplitude and phase estimation (APES) approach [11], [12]. Although a relatively straightforward analysis shows that (under certain conditions) both APES and Capon are essentially equivalent to the DFT as the amount of data goes to infinity, their finite sample properties have shown to be promising in experiments with both simulated and measured data. There is also some evidence that these methods may be able to improve the performance of automatic target recognition algorithms [13]. Therefore, several research papers describing these filterbank methods, as well as various extensions and combinations of them, have appeared in the literature. The main goal of this chapter is to summarize this body of research in a unified framework. We will also provide numerical illustrations of several of the methods, as well as identify some open problems.

We start in Section 4.2 by discussing the spectral estimation problem in general and by introducing appropriate notation. In Section 4.3 we go on to discuss several spectral estimators that can be obtained via a weighted least-squares (WLS) fitting criterion. These methods include in particular the smoothed DFT, Capon, APES and their variations. In Section 4.4, we provide a filterbank interpretation of the so-obtained estimators. Section 4.5 shows how the estimators can be extended to two-dimensional data, and Section 4.6 discusses how the so-called forward-backward averaging can be used to improve the statistical properties of the estimators. In Section 4.7, we summarize the available results on statistical performance of adaptive filterbank estimators. Next, in Section 4.8 we describe fast techniques for the evaluation of the Capon and APES spectra and in Section 4.9 we collect various numerical examples of the performance of these estimators. Section 4.10 treats an important problem where the filterbank estimators have proven to be successful: spectral estimation of data with missing samples. Finally, in Section 4.11 we treat some special topics including spectral estimation of real-valued data, spectral analysis of data that contain damped sinusoids in noise, and combinations of the various filterbank methods.

4.2 The Spectral Estimation Problem

In a few words, the spectral estimation problem amounts to “estimating how the total power is distributed over frequency, from a finite record of a stationary data sequence.” To formulate this statement in a more precise form, it is necessary to first specify what we mean by “distribution of power over frequency.”

There is more than one way to define the spectral distribution of power. For instance, for a stationary random process $x(n)$ we can define the *power spectrum* [14]:

$$\alpha_P(\omega) \triangleq \sum_{k=-\infty}^{\infty} E[x(n)x^*(n-k)]e^{-i\omega k} \quad (4.2.1)$$

where $(\cdot)^*$ stands for the complex conjugate. On the other hand, if $x(n)$ is a signal consisting of superimposed sinusoids with fixed amplitudes in stationary noise $w(n)$ that has a finite power spectrum for all ω :

$$x(n) = \sum_{p=1}^P \alpha_p e^{i\omega_p n} + w(n), \quad (4.2.2)$$

it is natural to define the *complex amplitude spectrum* in the following way:¹

$$\alpha_A(\omega) \triangleq \begin{cases} \alpha_p & \omega = \omega_p \\ 0 & \text{otherwise} \end{cases}. \quad (4.2.3)$$

For a signal such as that in (4.2.2), the power spectrum as defined in (4.2.1) is infinite whenever ω is equal to any of the frequencies of the sinusoids. Moreover, if the signal amplitudes $\{\alpha_k\}$ are fixed, the signal in (4.2.2) is *not* stationary (it is cyclo-stationary [14]).

Before the spectral estimation problem is formulated, it is necessary to specify what type of signal that will be studied, and how the spectrum that we want to estimate is defined. Throughout this chapter, and often elsewhere in the context of nonparametric spectral estimation, it is assumed that the signal under consideration is a sum of sinusoids embedded in stationary noise, i.e., that (4.2.2) holds, and consequently that the amplitude spectrum in (4.2.3) is the quantity of interest. On the other hand, signals observed in practice may not necessarily consist of superimposed sinusoids; if the signal is not a superposition of sinusoids then the estimation of the amplitude spectrum apparently amounts to implicitly *approximating* the signal with such a sum of sinusoids. Doing so may make sense in a wide range of applications. For instance, the presence of a hidden periodicity in the data can be revealed by estimating the amplitude of a fictitious sinusoidal component in the data and comparing it to a threshold. In SAR imaging applications, the measured phase-history data fulfill a two-dimensional counterpart to the model (4.2.2) (in an approximate sense).

¹For deterministic signals with finite energy, there are also other definitions of the amplitude spectrum, see, e.g., [7, Sec. 2.5].

4.3 Nonparametric Spectral Analysis via Weighted Least-Squares

In this section we introduce the nonparametric spectral analysis methods. We first describe these methods in a weighted least-squares (WLS) framework, and later in Section 4.4 we discuss filterbank interpretations of them.

Let $x(n)$ be an arbitrary one-dimensional (1D) time series. For a given frequency ω of interest, let us decompose $x(n)$ in a sinusoidal term with amplitude $\alpha(\omega)$ and a residual term $w_\omega(n)$:

$$x(n) = \alpha(\omega)e^{i\omega n} + w_\omega(n). \quad (4.3.1)$$

In the context of (4.2.2), $w_\omega(n)$ contains all sinusoids except for that one possibly present at ω , as well as the random noise component $w(n)$. The problem of estimating the amplitude spectrum of $x(n)$ then essentially becomes that of estimating $\alpha(\omega)$ given $x(n)$.

Let us assume that we are given N consecutive samples of $x(n)$, viz., $x(n)$ for $n = 0, \dots, N-1$. Let $M \leq N/2$ be a user parameter, whose choice will be discussed subsequently, and arrange the observed data $x(n)$ in an $M \times L$ Hankel matrix as follows:

$$\mathbf{Y} \triangleq [\mathbf{y}_0 \cdots \mathbf{y}_{L-1}] \triangleq \begin{bmatrix} x(0) & x(1) & \cdots & x(L-2) & x(L-1) \\ x(1) & x(2) & & x(L-1) & x(L) \\ \vdots & & \ddots & & \vdots \\ x(M-2) & x(M-1) & & x(N-3) & x(N-2) \\ x(M-1) & x(M) & \cdots & x(N-2) & x(N-1) \end{bmatrix} \quad (4.3.2)$$

where

$$L = N - M + 1. \quad (4.3.3)$$

We will refer to the columns $\{\mathbf{y}_l\}$ of \mathbf{Y} as *snapshots*. Also, let

$$\mathbf{W}(\omega) = [\mathbf{w}_0(\omega) \cdots \mathbf{w}_{L-1}(\omega)] \quad (4.3.4)$$

be defined similarly to \mathbf{Y} in (4.3.2) and let $\mathbf{A}(\omega)$ be the following rank-one matrix:

$$\mathbf{A}(\omega) = \mathbf{a}_M(\omega)\mathbf{a}_L^T(\omega) \quad (4.3.5)$$

where $(\cdot)^T$ denotes the transpose and

$$\mathbf{a}_P(\omega) \triangleq \begin{bmatrix} 1 \\ e^{i\omega} \\ \vdots \\ e^{i\omega(P-1)} \end{bmatrix} \quad (4.3.6)$$

for any integer P . Then we can write (4.3.1) as:

$$\mathbf{Y} = \alpha(\omega)\mathbf{A}(\omega) + \mathbf{W}(\omega). \quad (4.3.7)$$

Note that in the “degenerate” case of $M = 1$, (4.3.7) reduces to

$$\begin{aligned} & [x(0) \quad \cdots \quad x(N-1)] \\ = & \alpha(\omega) [1 \quad e^{i\omega} \quad \cdots \quad e^{i\omega(N-1)}] + [w_\omega(0) \quad \cdots \quad w_\omega(N-1)]. \end{aligned} \quad (4.3.8)$$

Equation (4.3.7) expresses a linear regression between the quantity of interest $\alpha(\omega)$ and the observed data matrix \mathbf{Y} , and is similar to the classical snapshot formulation that appears in most treatments of spectral estimation methods [5], [6], [7]. For instance, all the so-called parametric subspace methods [5], [9] rely on an equation equivalent to (4.3.7), which in that case is essential in order to formulate a low-rank signal model. The DFT method, certain averaged DFT methods, as well as the Capon and APES algorithms are also based on (4.3.7). In the present framework, the latter methods simply compute a WLS estimate of $\alpha(\omega)$ as follows:

$$\hat{\alpha}(\omega) = \underset{\alpha(\omega)}{\operatorname{argmin}} \left\| \Phi^{-1/2}(\omega)(\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega)) \right\|^2 \quad (4.3.9)$$

where $\Phi(\omega)$ is a weighting matrix and $(\cdot)^{1/2}$ denotes the square root of a Hermitian matrix.

The WLS criterion (4.3.9) effectively pre-whitens the columns of \mathbf{Y} with an ω -dependent weight matrix $\Phi(\omega)$ to account for a possible coloration of the residual signal $w_\omega(n)$, and then fits a single-sinusoidal model to the pre-whitened data. Provided that $w_\omega(n)$ is stationary, the residual vectors $\{\mathbf{w}_l(\omega)\}$ are identically distributed, although they are *not* independent. Consequently, it makes sense to use the same weight matrix $\Phi(\omega)$ to pre-whiten all columns of \mathbf{Y} .

The criterion in (4.3.9) is quadratic in $\alpha(\omega)$. Hence the solution to (4.3.9) can be obtained by some straightforward algebra:

$$\hat{\alpha}(\omega) = \frac{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega)} \quad (4.3.10)$$

where $(\cdot)^H$ denotes the complex conjugate transpose and

$$\mathbf{g}(\omega) \triangleq \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}_l e^{-i\omega l} = \frac{1}{L} \mathbf{Y} \mathbf{a}_L^*(\omega). \quad (4.3.11)$$

The difference between the DFT, Capon and APES methods is the choice of M and $\Phi(\omega)$, which will be discussed next.

However, before we proceed to discuss the different methods, a technical remark is in order. For many of the estimates presented below, the matrix $\Phi(\omega)$ is a function

of the covariance sequence of $x(n)$. In a strict sense, this covariance sequence is only well-defined if $x(n)$ is a stationary random process. This is the case if $x(n)$ is an outcome of (4.2.2) and the amplitudes $\{\alpha_p\}$ are considered to be *random* quantities, but *not* if $x(n)$ is a realization of (4.2.2) *conditioned* on $\{\alpha_p\}$ (nevertheless, when applying the filterbank estimators we consider $\{\alpha_p\}$ to be deterministic). Under the assumption that $x(n)$ is stationary, the following covariance matrix is well-defined:

$$\mathbf{R} \triangleq E[\mathbf{y}_k \mathbf{y}_k^H]. \quad (4.3.12)$$

(and is independent of k). Moreover, since $\{\mathbf{w}_k(\omega)\}$ are identically distributed, the covariance matrix

$$\mathbf{Q}(\omega) \triangleq E[\mathbf{w}_k(\omega) \mathbf{w}_k^H(\omega)] \quad (4.3.13)$$

is also well-defined (and independent of k). Finally, under the same assumption, it follows readily that

$$\mathbf{Q}(\omega) = \mathbf{R} - E[|\alpha(\omega)|^2] \mathbf{a}_M(\omega) \mathbf{a}_M^H(\omega). \quad (4.3.14)$$

4.3.1 The Discrete Fourier Transform (DFT) Method

For the DFT method, we take $M = 1$ and consequently we have $L = N$. In this case, $\Phi(\omega)$ disappears from both the formulation (4.3.9) and from the solution (4.3.10) of the LS problem. The resulting spectral estimate (4.3.10) reduces to the well-known DFT estimate:

$$\hat{\alpha}_{\text{DFT}}(\omega) = \frac{1}{N} \mathbf{a}_N^H(\omega) \mathbf{y} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-i\omega n} \quad (4.3.15)$$

where

$$\mathbf{y} = [x(0) \ \cdots \ x(N-1)]^T. \quad (4.3.16)$$

4.3.2 The Averaged Fourier Method

If we choose $M > 1$ and $\Phi(\omega) = \mathbf{I}$ (independently of ω), we obtain the following averaged version of the DFT:

$$\hat{\alpha}_{\text{A-DFT}}(\omega) = \frac{1}{M} \mathbf{a}_M^H(\omega) \mathbf{g}(\omega) = \frac{1}{LM} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x(l+m) e^{-i\omega(l+m)}. \quad (4.3.17)$$

The averaging in (4.3.17) reduces the variance of the spectral estimate at the cost of a decreased resolution. Effectively, in this case the parameter M can be interpreted as a smoothing lag that trades spectral resolution for improved statistical stability. It is worth mentioning that the averaged Fourier Method coincides with a version of the Welch method [15], [5].

4.3.3 The Capon Method

The Capon method [10], [5], [7], [6] is a classical spectral analysis method that has also, somewhat incorrectly, been referred to as a “maximum-likelihood” spectral estimator. The Capon method can easily be described in the current framework by choosing an ω -independent weighting matrix $\Phi(\omega)$ that is a sample estimate of the data covariance matrix \mathbf{R} in (4.3.12):

$$\Phi_{\text{Capon}}(\omega) = \hat{\mathbf{R}} \triangleq \frac{1}{L} \mathbf{Y} \mathbf{Y}^H = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}_l \mathbf{y}_l^H. \quad (4.3.18)$$

The resulting estimator is (cf. (4.3.10))

$$\hat{\alpha}_{\text{Capon}}(\omega) = \frac{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{a}_M(\omega)}. \quad (4.3.19)$$

The choice of $\Phi(\omega)$ in (4.3.18) can be given a natural interpretation as follows. Let $\mathbf{Q}(\omega)$ be the covariance matrix defined in (4.3.13). This matrix effectively captures the statistics of the residual term $w_\omega(n)$ in (4.3.1). An ideal choice of weighting matrix for the WLS problem would be to use $\Phi(\omega) = \mathbf{Q}(\omega)$ since this yields a Markov-like estimate that is known to have optimal (in a certain sense) statistical properties.² In practice \mathbf{Q} is unknown. However, under the assumptions made on the data we have that (see (4.3.14)):

$$\mathbf{R} = \mathbf{Q}(\omega) + E[|\alpha(\omega)|^2] \mathbf{a}_M(\omega) \mathbf{a}_M^H(\omega). \quad (4.3.20)$$

By applying the matrix inversion lemma [17] to (4.3.20), it can be shown that

$$\mathbf{R}^{-1} \mathbf{a}_M(\omega) = \frac{1}{1 + E[|\alpha|^2] \cdot \mathbf{a}_M^H(\omega) \mathbf{Q}^{-1}(\omega) \mathbf{a}_M(\omega)} \mathbf{Q}^{-1}(\omega) \mathbf{a}_M(\omega). \quad (4.3.21)$$

Hence we see from (4.3.10) that using $\Phi(\omega) = \mathbf{R}^{-1}$ yields the same result as using $\Phi(\omega) = \mathbf{Q}^{-1}(\omega)$. Although such a result does not hold when the matrices in (4.3.21) are replaced by their sample estimates, we can still expect that Capon’s choice of $\Phi(\omega) = \hat{\mathbf{R}}^{-1}$ will give estimates with reasonable properties.

In some literature, the “Capon” estimator derived above is referred to as the “amplitude-spectrum Capon (ASC) estimator.” There exists a variant of ASC, namely the power-spectrum Capon (PSC) estimator, which can also be derived in a framework similar to that above. Although for conciseness we do not treat PSC in this chapter, for instance [18], [19], [20] have observed that ASC and PSC have quite different properties (in terms of spectral resolution, bias and variance).

²However, even if $\mathbf{Q}(\omega)$ were available, the choice $\Phi(\omega) = \mathbf{Q}(\omega)$ would not be overall optimal since the columns of \mathbf{W} are not independent. This problem was addressed in [16]; see also the section “MMSE spectral estimation” below.

4.3.4 The APES Method

The APES algorithm was originally suggested in [11] and later re-derived and interpreted in different ways in [12], [21], [22], [23]. APES chooses $\Phi(\omega)$ as follows:

$$\Phi_{\text{APES}}(\omega) = \hat{\mathbf{R}} - \mathbf{g}(\omega)\mathbf{g}^H(\omega) \triangleq \hat{\mathbf{Q}}(\omega) \quad (4.3.22)$$

where $\mathbf{g}(\omega)$ is defined in (4.3.11). Note that unlike Capon, for APES the weighting matrix $\Phi(\omega)$ depends on ω . The resulting spectral estimate is obtained by inserting (4.3.22) into (4.3.10):

$$\hat{\alpha}_{\text{APES}}(\omega) = \frac{\mathbf{a}_M^H(\omega)\hat{\mathbf{Q}}^{-1}(\omega)\mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega)\hat{\mathbf{Q}}^{-1}(\omega)\mathbf{a}_M(\omega)}. \quad (4.3.23)$$

We have that (cf. (4.3.14)):

$$E[\hat{\mathbf{Q}}(\omega)] = \mathbf{R} - E[\mathbf{g}(\omega)\mathbf{g}^H(\omega)] \rightarrow \mathbf{R} - E[|\alpha(\omega)|^2]\mathbf{a}(\omega)\mathbf{a}^H(\omega) = \mathbf{Q}(\omega) \quad (4.3.24)$$

when $N \rightarrow \infty$. Hence $\hat{\mathbf{Q}}(\omega)$ can be seen as an estimate of $\mathbf{Q}(\omega)$ in (4.3.13), which makes the choice of $\Phi(\omega)$ in (4.3.22) intuitively appealing.

In [11] it was also shown that (4.3.23) is the maximum-likelihood estimate of $\alpha(\omega)$ in the model (4.3.7) under the assumption that the columns of \mathbf{W} are *independent* and Gaussian with the same covariance matrix. Unfortunately, the columns of \mathbf{W} are not independent for the problem under study; nevertheless this result does establish a connection between the seemingly ad-hoc choice of $\Phi(\omega)$ in (4.3.22) and a more systematic estimation theory. To prove that $\alpha(\omega)$ is the ML estimate in the aforementioned sense, assume that \mathbf{W} has independent and Gaussian columns with covariance matrix $\mathbf{Q}(\omega)$. Then the part of the negative log-likelihood function that depends on the quantities of interest is proportional to:

$$L \cdot \log |\Upsilon(\omega)| + \text{tr} \left[\Upsilon(\omega)^{-1} (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega)) (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega))^H \right]. \quad (4.3.25)$$

Applying some matrix algebra we find that the matrix $\mathbf{Q}(\omega)$ that minimizes (4.3.25) is equal to:

$$\mathbf{Q}_{\text{ML}}(\omega) = \frac{1}{L} (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega)) (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega))^H. \quad (4.3.26)$$

Inserting (4.3.26) into (4.3.25) we find that the problem of ML estimation of $\alpha(\omega)$ reduces to that of minimizing the following determinant:

$$\begin{aligned} & \left| (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega)) (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega))^H \right| \\ &= \left| (\mathbf{Y} - \alpha(\omega)\mathbf{a}_M(\omega)\mathbf{a}_L^T(\omega)) (\mathbf{Y} - \alpha(\omega)\mathbf{a}_M(\omega)\mathbf{a}_L^T(\omega))^H \right|. \end{aligned} \quad (4.3.27)$$

By using some identities involving the determinant of a matrix we obtain:

$$\begin{aligned}
& \left| (\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega))(\mathbf{Y} - \alpha(\omega)\mathbf{A}(\omega))^H \right| \\
&= \left| \mathbf{Y}\mathbf{Y}^H - \alpha(\omega)\mathbf{A}(\omega)\mathbf{Y}^H - \alpha^H(\omega)\mathbf{Y}\mathbf{A}^H(\omega) + |\alpha(\omega)|^2\mathbf{A}(\omega)\mathbf{A}^H(\omega) \right| \\
&= \left| \mathbf{Y}\mathbf{Y}^H - \mathbf{g}(\omega)\mathbf{g}(\omega) + (\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega))(\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega))^H \right| \\
&= \left| \mathbf{Y}\mathbf{Y}^H - \mathbf{g}(\omega)\mathbf{g}(\omega) \right| \cdot \left| \mathbf{I} + \hat{\mathbf{Q}}^{-1}(\omega)(\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega))(\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega))^H \right| \\
&= \left| \mathbf{Y}\mathbf{Y}^H - \mathbf{g}(\omega)\mathbf{g}(\omega) \right| \cdot \left(1 + (\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega))^H \hat{\mathbf{Q}}^{-1}(\omega)(\alpha(\omega)\mathbf{a}_M(\omega) - \mathbf{g}(\omega)) \right)
\end{aligned} \tag{4.3.28}$$

where $\hat{\mathbf{Q}}(\omega)$ is taken as in (4.3.22). The expression in the last line of (4.3.28) is quadratic in $\alpha(\omega)$ and is minimized by (4.3.23).

4.3.5 MMSE Spectral Estimation

A different approach to the spectral analysis problem, which can be argued to be more systematic than both Capon and APES, was proposed in [16]. By developing an expression for the mean-square error (MSE) of the WLS spectral estimate (4.3.10), an optimality criterion for the choice of $\Phi(\omega)$ by minimizing this MSE was formulated. In the current framework, the results of [16] can be described as follows. Since

$$\hat{\alpha}(\omega) - \alpha(\omega) = \frac{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)(\mathbf{g}(\omega) - \alpha(\omega)\mathbf{a}_M(\omega))}{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega)} \tag{4.3.29}$$

we find by using the fact that (cf. (4.3.11)):

$$\mathbf{g}(\omega) - \alpha(\omega)\mathbf{a}_M(\omega) = \frac{1}{L}\mathbf{W}\mathbf{a}_L^*(\omega) \tag{4.3.30}$$

that the MSE for the estimate of $\alpha(\omega)$ in (4.3.29) is equal to:

$$\begin{aligned}
E[|\hat{\alpha}(\omega) - \alpha(\omega)|^2] &= \frac{1}{L^2} E \left[\frac{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{W}\mathbf{a}_L^*(\omega)\mathbf{a}_L^T(\omega)\mathbf{W}^H\Phi^{-1}(\omega)\mathbf{a}_M(\omega)}{(\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega))^2} \right] \\
&= \frac{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\Upsilon(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega)}{(\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega))^2}
\end{aligned} \tag{4.3.31}$$

where

$$\Upsilon(\omega) = \frac{1}{L^2} E[\mathbf{W}\mathbf{a}_L^*(\omega)\mathbf{a}_L^T(\omega)\mathbf{W}^H]. \tag{4.3.32}$$

We have that:

$$\Upsilon(\omega) = \frac{1}{L^2} E[\mathbf{Y}\mathbf{a}_L^*(\omega)\mathbf{a}_L^T(\omega)\mathbf{Y}^H] - E[|\alpha(\omega)|^2]\mathbf{a}_M(\omega)\mathbf{a}_M^H(\omega). \tag{4.3.33}$$

It follows that the MSE in (4.3.31) is minimized for

$$\Phi(\omega) = E[\mathbf{Y}\mathbf{a}_L^*(\omega)\mathbf{a}_L^T(\omega)\mathbf{Y}^H] \triangleq \Psi(\omega). \quad (4.3.34)$$

The problem of spectral estimation in a MMSE sense therefore reduces to that of finding a good estimate of $\Psi(\omega)$. This problem was addressed in some detail in [16], where it was suggested that $\Psi(\omega)$ could be estimated as a structured quadratic function of the data (this is quite natural in view of (4.3.34)):

$$\hat{\Psi}(\omega) = \mathbf{Y}\Sigma(\omega)\mathbf{Y}^H \quad (4.3.35)$$

where $\Sigma(\omega)$ is an $L \times L$ Toeplitz matrix whose (k, l) th element is equal to

$$\Sigma_{k,l}(\omega) = r_{k-l}e^{-i\omega(k-l)} \quad (4.3.36)$$

and $\{r_k\}$ is a conjugate-symmetric weighting sequence. A motivation for choosing this estimate that relates to the literature on estimation of covariance sequences was provided in [16]. As a guideline towards choosing the constants $\{r_l\}$, a theorem in the cited article demonstrated that as long as

$$f(\gamma) = \sum_{l=-L+1}^{L-1} r_l e^{i\gamma l} \quad (4.3.37)$$

is uniformly (in γ) lower bounded by a strictly positive constant, the condition number of the matrix $\Psi(\omega)$ is bounded, which is a desirable property. Yet, choosing the “best” windowing sequence $\{r_l\}$ remains an open problem. Numerical experiments reported in [16] indicate that the MMSE estimator can provide slightly better resolution than APES and Capon, but we will not pursue it further in this chapter.

4.4 Matched-Filterbank Interpretations

In addition to the averaged Fourier transform, which can be seen as a degenerated special case of the filterbank methods, perhaps the most successful estimators to date are the Capon and APES methods; hence the rest of this chapter will focus on these methods.

The DFT, Capon and APES estimators described above can be interpreted in an *adaptive filterbank* framework. Although this interpretation is somewhat ad-hoc, it does provide additional insight into the characteristics of the different methods. The filterbank interpretations of the various estimators was presented in [23]. In the context of filterbanks, for each frequency ω of interest the nonparametric methods filter the data with a normalized finite-impulse response (FIR) filter $\mathbf{h}(\omega)$ of length M . The variable M is a user parameter which is the same as the length of the snapshots in the matrix (4.3.2) above. The filter $\mathbf{h}(\omega)$ is chosen according to a criterion which is different for the various spectral analysis methods, but with the common constraint that a sinusoid with frequency ω should pass the filter without

distortion. Following the filtering, a sinusoid is fitted to the filtered data in a LS sense, and the amplitude of the so-obtained sinusoid $\hat{\alpha}(\omega)$ is taken as the estimate of the amplitude spectrum $\alpha_A(\omega)$ at the frequency ω of interest.

The idea behind the filtering is easy to understand intuitively as follows. Clearly, filtering $x(n)$ with $\mathbf{h}(\omega)$ reduces the number of available samples from N to L ; yet, if the filter is properly designed the increase in signal-to-noise ratio (SNR) in the filtered data will counterbalance this reduction and the amplitude of a sinusoidal component with frequency ω can be estimated more accurately from the filtered data than from the original signal.

Let $x(n)$ be the same time-series as above, and let \mathbf{y}_n be defined according to (4.3.2). If

$$\mathbf{h}(\omega) = [h_0(\omega) \cdots h_{M-1}(\omega)]^T \quad (4.4.1)$$

is an M -vector that contains the coefficients of the FIR filter, the filtered data can be written as:

$$z(l) = \sum_{m=0}^{M-1} x(l+m)h_m(\omega) = \mathbf{h}^H(\omega)\mathbf{y}_l = \alpha(\omega)\mathbf{h}^H(\omega)\mathbf{a}_M(\omega) + \mathbf{h}^H(\omega)\mathbf{w}_l(\omega) \quad (4.4.2)$$

for $l = 0, \dots, L-1$. If the transfer function of the filter is equal to unity at ω :

$$\mathbf{h}^H(\omega)\mathbf{a}_M(\omega) = 1 \quad (4.4.3)$$

(in other words, a sinusoid with frequency ω passes the filter undistorted), then the LS estimate of $\alpha(\omega)$ (given \mathbf{y}_l) in (4.4.2) is:

$$\hat{\alpha}(\omega) = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{h}^H(\omega)\mathbf{y}_l e^{-i\omega l} = \mathbf{h}^H(\omega)\mathbf{g}(\omega) \quad (4.4.4)$$

where $\mathbf{h}(\omega)$ is yet to be determined.

4.4.1 DFT and Averaged DFT

The averaged DFT can be interpreted in a filterbank framework by choosing a fixed (i.e. independent of the realization of $x(n)$) narrowband filter $\mathbf{h}(\omega)$ as follows:

$$\mathbf{h}(\omega) = \frac{1}{M} [1 e^{i\omega} \cdots e^{i\omega(M-1)}]^T. \quad (4.4.5)$$

We get the spectral estimate

$$\begin{aligned} \hat{\alpha}_{\text{A-DFT}}(\omega) &= \frac{1}{L} \sum_{t=0}^{L-1} \frac{1}{M} \sum_{k=0}^{M-1} x(t+k) e^{-i\omega(t+k)} \\ &= \frac{1}{L} \sum_{t=0}^{L-1} e^{-i\omega t} \left(\frac{1}{M} \sum_{k=0}^{M-1} x(t+k) e^{-i\omega k} \right) \end{aligned} \quad (4.4.6)$$

which is (4.3.17).

4.4.2 Capon

For the Capon method, the filter coefficients in $\mathbf{h}(\omega)$ are chosen such that the energy of the filtered signal is minimized subject to the constraint that a sinusoid with frequency ω passes the filter undistorted. This design criterion can be written as

$$\min_{\mathbf{h}(\omega)} \sum_{l=0}^{L-1} |\mathbf{h}^H(\omega) \mathbf{y}_l|^2 \quad (4.4.7)$$

subject to the constraint

$$\mathbf{a}_M^H(\omega) \mathbf{h}(\omega) = 1. \quad (4.4.8)$$

The constraint (4.4.8) is enforced in order to guarantee the existence of a meaningful solution to (4.4.7), and it is also intuitively appealing. The solution to the minimization problem under consideration is:

$$\mathbf{h}(\omega) = \frac{\hat{\mathbf{R}}^{-1} \mathbf{a}_M(\omega)}{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{a}_M(\omega)}. \quad (4.4.9)$$

The Capon amplitude spectrum estimate $\hat{\alpha}(\omega)$ is obtained by fitting a sinusoid to the filtered data $z(l) = \mathbf{h}^H(\omega) \mathbf{y}_l$ in a least-squares (LS) sense. This yields the same estimate as previously derived in a different way, viz. (4.3.19):

$$\hat{\alpha}_{\text{Capon}}(\omega) = \frac{1}{L} \sum_{l=0}^{L-1} y(l) e^{-i\omega l} = \frac{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{a}_M(\omega)}. \quad (4.4.10)$$

4.4.3 APES

For the APES method the filter $\mathbf{h}(\omega)$ and the amplitude spectrum estimate $\hat{\alpha}(\omega)$ are the joint minimizers of the following criterion:

$$\min_{\mathbf{h}(\omega), \alpha(\omega)} \sum_{l=0}^{L-1} |\mathbf{h}^H(\omega) \mathbf{y}_l - \alpha(\omega) e^{i\omega l}|^2 \quad (4.4.11)$$

subject to the same constraint (4.4.8) as for the Capon method. The design in (4.4.11) can be interpreted as filtering $x(n)$ in a way such that the filtered signal resembles a sinusoid as closely as possible in a least-squares (LS) sense; this is an intuitively appealing criterion. The spectral estimate resulting from (4.4.11) is equal to that derived in (4.3.23):

$$\hat{\alpha}_{\text{Apes}}(\omega) = \frac{\mathbf{a}_M^H(\omega) \hat{\mathbf{Q}}^{-1}(\omega) \mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega) \hat{\mathbf{Q}}^{-1}(\omega) \mathbf{a}_M(\omega)}. \quad (4.4.12)$$

4.5 Extensions to Two-Dimensional Data

With some additional notational effort, the nonparametric spectral estimation methods can easily be extended to spectral estimation of two-dimensional (2D) data. Let $X(n, \bar{n})$ be a 2D data matrix of dimension $N \times \bar{N}$, and suppose that for a certain frequency pair $\omega, \bar{\omega}$ of interest we decompose $X(n, \bar{n})$ in a manner similar to (4.3.7):

$$X(n, \bar{n}) = \alpha(\omega, \bar{\omega})e^{i(\omega n + \bar{\omega} \bar{n})} + W_{\omega, \bar{\omega}}(n, \bar{n}). \quad (4.5.1)$$

Also, let M and \bar{M} be user parameters and let us define snapshots that are vectorized sub-matrices of the available data matrix:

$$\mathbf{y}_{l, \bar{l}} \triangleq \text{vec} \left(\begin{bmatrix} X(l, \bar{l}) & \cdots & X(l, \bar{l} + \bar{M} - 1) \\ \vdots & \ddots & \vdots \\ X(l + M - 1, \bar{l}) & \cdots & X(l + M - 1, \bar{l} + \bar{M} - 1) \end{bmatrix} \right) \quad (4.5.2)$$

for $l = 0, \dots, L - 1$, $\bar{l} = 0, \dots, \bar{L} - 1$ where $L = N - M + 1$ and $\bar{L} = \bar{N} - \bar{M} + 1$ (note that in the 1D case we have $N = M = L = 1$).

Let us introduce the following 2D Fourier vector, for arbitrary integers P and \bar{P} :

$$\mathbf{a}_{P, \bar{P}}(\omega, \bar{\omega}) \triangleq [1 \ e^{i\bar{\omega}} \ \cdots \ e^{i(\bar{P}-1)\bar{\omega}}]^T \otimes [1 \ e^{i\omega} \ \cdots \ e^{i(P-1)\omega}]^T \quad (4.5.3)$$

where \otimes stands for the Kronecker product and let

$$\mathbf{Y} \triangleq [\mathbf{y}_{0,0} \ \cdots \ \mathbf{y}_{L-1,0} \ \mathbf{y}_{0,1} \ \cdots \ \mathbf{y}_{L-1,\bar{L}-1}] \quad (4.5.4)$$

as well as

$$\mathbf{A}(\omega, \bar{\omega}) = \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}^T(\omega, \bar{\omega}). \quad (4.5.5)$$

Then we can formulate a linear regression similar to (4.3.7):

$$\mathbf{Y} = \alpha(\omega, \bar{\omega}) \mathbf{A}(\omega, \bar{\omega}) + \mathbf{W}(\omega, \bar{\omega}). \quad (4.5.6)$$

Estimation of $\alpha(\omega, \bar{\omega})$ in (4.5.6) via WLS, with $\Phi(\omega, \bar{\omega})$ as weighting matrix, yields an expression with the same form as (4.3.10):

$$\hat{\alpha}(\omega, \bar{\omega}) = \frac{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \Phi^{-1}(\omega, \bar{\omega}) \mathbf{g}(\omega, \bar{\omega})}{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \Phi^{-1}(\omega, \bar{\omega}) \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega})} \quad (4.5.7)$$

where

$$\mathbf{g}(\omega, \bar{\omega}) \triangleq \frac{1}{LL} \mathbf{Y} \mathbf{a}_{L, \bar{L}}^*(\omega, \bar{\omega}) = \frac{1}{LL} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l, \bar{l}} e^{-i(\omega l + \bar{\omega} \bar{l})} \quad (4.5.8)$$

where \mathbf{Y} is defined in (4.5.4), and $\Phi(\omega, \bar{\omega})$ is chosen in a manner similar to that in Section 4.3.

Capon: For the Capon method, $\Phi(\omega, \bar{\omega})$ is chosen according to:

$$\Phi = \hat{\mathbf{R}} = \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l,\bar{l}} \mathbf{y}_{l,\bar{l}}^H = \frac{1}{L\bar{L}} \mathbf{Y} \mathbf{Y}^H \quad (4.5.9)$$

(independently of $\omega, \bar{\omega}$).

APES: For the APES algorithm, $\Phi(\omega, \bar{\omega})$ is taken as:

$$\Phi(\omega, \bar{\omega}) = \hat{\mathbf{Q}}(\omega, \bar{\omega}) = \hat{\mathbf{R}} - \mathbf{g}(\omega, \bar{\omega}) \mathbf{g}^H(\omega, \bar{\omega}). \quad (4.5.10)$$

4.5.1 Representation of Two-dimensional Filterbanks

In this section we introduce some notation for 2D filtering, that will help us formulate the nonparametric spectral estimators for 2D data in a filterbank framework.

Let \mathbf{X} be an arbitrary $N \times \bar{N}$ data matrix and let \mathbf{H} be an $M \times \bar{M}$ matrix that contains the coefficients of a given FIR filter. Then we denote by

$$\mathbf{Z} = \mathbf{H} * \mathbf{X} \quad (4.5.11)$$

the following $L \times \bar{L}$ data matrix obtained by filtering \mathbf{X} through the filter corresponding to \mathbf{H} :

$$Z(l, \bar{l}) = \sum_{m=0}^{M-1} \sum_{\bar{m}=0}^{\bar{M}-1} H^*(m, \bar{m}) X(l+m, \bar{l}+\bar{m}) = \text{vec}^H(\mathbf{H}) \mathbf{y}_{l,\bar{l}}. \quad (4.5.12)$$

In (4.5.12), $\mathbf{y}_{l,\bar{l}}$ is defined as in (4.5.2). As the size of the filtered data matrix is $L \times \bar{L}$ there are no edge-effects.

Let the $L\bar{L} \times (\bar{L}N - M + 1)$ matrix \mathbf{T} be defined by:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_L & \mathbf{0}_{L,M-1} & & & & & & & & \\ & & \mathbf{I}_L & \mathbf{0}_{L,M-1} & & & & & & \\ & & & & \ddots & & & & & \\ & & & & & & & & & \mathbf{I}_L \end{bmatrix} \quad (4.5.13)$$

where \mathbf{I}_L is the $L \times L$ identity matrix and $\mathbf{0}_{K \times L}$ is a $K \times L$ matrix of zeros. Note that in the 1D-case, \mathbf{T} is an identity matrix. Furthermore, let \mathbf{G} be the following Toeplitz matrix of dimension $(\bar{L}N - M + 1) \times N\bar{N}$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{h}_1^H & \mathbf{0}_{1,L-1} & \mathbf{h}_2^H & \mathbf{0}_{1,L-1} & \cdots & \mathbf{h}_M^H & 0 & \cdots & 0 \\ 0 & \mathbf{h}_1^H & \mathbf{0}_{1,L-1} & \mathbf{h}_2^H & \mathbf{0}_{1,L-1} & \cdots & \mathbf{h}_M^H & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{h}_1^H & \mathbf{0}_{1,L-1} & \mathbf{h}_2^H & \mathbf{0}_{1,L-1} & \cdots & \mathbf{h}_M^H \end{bmatrix} \quad (4.5.14)$$

where $\{\mathbf{h}_m\}_{m=1}^{\bar{M}}$ are the columns of \mathbf{H} . Note that in the 1D-case, \mathbf{H} is a row vector and hence \mathbf{G} is simply a Toeplitz matrix with \mathbf{H}^* as its first row. From (4.5.2), (4.5.12), (4.5.13) and (4.5.14) it follows after a few lines of algebra that

$$\text{vec}(\mathbf{Z}) = \text{vec}(\mathbf{H} * \mathbf{X}) = \mathbf{T}\mathbf{G} \text{vec}(\mathbf{X}). \quad (4.5.15)$$

Equation (4.5.15) will be useful later in Subsection 4.10.2.

4.5.2 Filterbank Interpretations of Two-Dimensional Spectral Estimators

Using the notation in the previous subsection, the filterbank interpretations of Capon and APES can easily be extended to the 2D case.

Capon: For Capon, the relevant criterion can be expressed as:

$$\min_{\mathbf{H}(\omega, \bar{\omega})} \|\mathbf{H}(\omega, \bar{\omega}) * \mathbf{X}\|^2 \quad (4.5.16)$$

subject to the constraint

$$\text{vec}^H(\mathbf{H}(\omega, \bar{\omega})) \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega}) = 1. \quad (4.5.17)$$

In the same way as for 1D data, the constraint in (4.5.17) guarantees that the transfer function of the filter $\mathbf{H}(\omega, \bar{\omega})$ is equal to unity at the frequency $\omega, \bar{\omega}$ and hence that a sinusoid with frequency $\omega, \bar{\omega}$ passes through the filter undistorted.

APES: For APES, the relevant criterion can be expressed as:

$$\min_{\mathbf{H}(\omega, \bar{\omega}), \alpha(\omega, \bar{\omega})} \|\text{vec}(\mathbf{H}(\omega, \bar{\omega}) * \mathbf{X}) - \alpha(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}(\omega, \bar{\omega})\|^2 \quad (4.5.18)$$

subject to the constraint (4.5.17).

4.6 Forward-Backward Averaging

Forward-backward (FB) averaging is a technique developed in the literature on parameter estimation [24], [25], [26]³, that can also be used to improve the performance of the adaptive filterbank estimators. The idea behind FB averaging is simple. Under the assumptions made in the preceding sections, the covariance matrices \mathbf{R} as well as $\mathbf{Q}(\omega)$ are Toeplitz matrices. In particular, this means that they are also persymmetric, that is, $\mathbf{R} = \mathbf{J}\mathbf{R}^T\mathbf{J}$ where \mathbf{J} is a reversal matrix (i.e., a matrix of compatible dimension that has ones along its anti-diagonal, and zeros everywhere else). Clearly, any attempt to estimate the matrices \mathbf{R} or $\mathbf{Q}(\omega)$ by forming (outer product) sample covariances, such as (4.3.18) and (4.3.22), leads to an estimate that is *not* Toeplitz. Incorporating the constraint that the covariance matrix is Toeplitz

³Reference [26] deals with 1D data, but we can expect the results to also be valid for 2D data.

leads to a difficult estimation problem (that is addressed in the literature on estimation of structured covariance matrices). It is easier to incorporate the constraint that the estimated covariance matrix should be persymmetric; this is done by FB averaging of the sample covariance matrix.

4.6.1 FB Averaging of the Covariance Matrices

The FB averaged spectral estimates take on the following form (for generality, we discuss FB averaging for the 2D case; the 1D case follows as a special instance). Let $\tilde{\mathbf{X}}$ be the following *backward* data matrix

$$\tilde{\mathbf{X}} \triangleq \begin{bmatrix} X^*(N-1, \bar{N}-1) & \cdots & X^*(N-1, 0) \\ \vdots & \ddots & \vdots \\ X^*(0, \bar{N}-1) & \cdots & X^*(0, 0) \end{bmatrix} \quad (4.6.1)$$

and let $\tilde{\mathbf{y}}_{l,\bar{l}}$ be defined similarly to $\mathbf{y}_{l,\bar{l}}$ by replacing $\mathbf{X}_{n,\bar{n}}$ with $\tilde{\mathbf{X}}_{n,\bar{n}}$ in (4.5.2). Also, let

$$\tilde{\mathbf{Y}} = \mathbf{J}\mathbf{Y}^*\mathbf{J} \quad (4.6.2)$$

where \mathbf{J} is a reversal matrix. Next, define the $M\bar{M} \times M\bar{M}$ *forward sample covariance matrix* (which is the same as before)

$$\hat{\mathbf{R}}_f \triangleq \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l,\bar{l}} \mathbf{y}_{l,\bar{l}}^H = \frac{1}{L\bar{L}} \mathbf{Y}\mathbf{Y}^H \quad (4.6.3)$$

where $\mathbf{y}_{l,\bar{l}}$ is defined in (4.5.2) and the *backward sample covariance matrix*

$$\hat{\mathbf{R}}_b \triangleq \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \tilde{\mathbf{y}}_{l,\bar{l}} \tilde{\mathbf{y}}_{l,\bar{l}}^H = \frac{1}{L\bar{L}} \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^H = \frac{1}{L\bar{L}} \mathbf{J}\mathbf{Y}^*\mathbf{Y}^T\mathbf{J} = \mathbf{J}\hat{\mathbf{R}}_f^T\mathbf{J}. \quad (4.6.4)$$

Then the *forward-backward* sample covariance matrix is defined according to:

$$\hat{\mathbf{R}}_{fb} \triangleq \frac{1}{2}(\hat{\mathbf{R}}_f + \hat{\mathbf{R}}_b). \quad (4.6.5)$$

Let us introduce the following vector-valued Fourier transforms:

$$\begin{aligned} \bar{\mathbf{g}}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \mathbf{y}_{l,\bar{l}} e^{-i(\omega l + \bar{\omega} \bar{l})} \\ \tilde{\mathbf{g}}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \tilde{\mathbf{y}}_{l,\bar{l}} e^{-i(\omega l + \bar{\omega} \bar{l})} \end{aligned} \quad (4.6.6)$$

and define the matrix

$$\begin{aligned}\hat{\mathbf{Q}}_{fb}(\omega, \bar{\omega}) &\triangleq \frac{1}{2}(\hat{\mathbf{Q}}(\omega, \bar{\omega}) + \mathbf{J}\hat{\mathbf{Q}}^T(\omega, \bar{\omega})\mathbf{J}) \\ &= \hat{\mathbf{R}}_{fb} - \frac{1}{2} \begin{bmatrix} \bar{\mathbf{g}}(\omega, \bar{\omega}) & \tilde{\mathbf{g}}(\omega, \bar{\omega}) \end{bmatrix} \begin{bmatrix} \bar{\mathbf{g}}^H(\omega, \bar{\omega}) \\ \tilde{\mathbf{g}}^H(\omega, \bar{\omega}) \end{bmatrix}.\end{aligned}\quad (4.6.7)$$

Then the Capon estimate of $\alpha_A(\omega, \bar{\omega})$ is

$$\hat{\alpha}_{\text{FB-Capon}}(\omega, \bar{\omega}) = \frac{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{R}}_{fb}^{-1} \bar{\mathbf{g}}(\omega, \bar{\omega})}{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega})} \quad (4.6.8)$$

and the APES estimate is

$$\hat{\alpha}_{\text{FB-APES}} = \frac{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{Q}}_{fb}^{-1}(\omega, \bar{\omega}) \bar{\mathbf{g}}(\omega, \bar{\omega})}{\mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{Q}}_{fb}^{-1}(\omega, \bar{\omega}) \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega})}. \quad (4.6.9)$$

4.6.2 Filterbank Interpretation of FB Averaged Spectral Estimators

In the context of adaptive filterbanks, the FB averaging can be accomplished by running the data forwards as well as backwards through the FIR filter, and rewriting the relevant criteria as follows.

Capon: The filter coefficient matrix $\mathbf{H}(\omega, \bar{\omega})$ associated with the FB-averaged Capon spectrum is the minimizer of the following criterion:

$$\min_{\mathbf{H}(\omega, \bar{\omega})} \left\{ \|\mathbf{H}(\omega, \bar{\omega}) * \mathbf{X}\|^2 + \|\mathbf{H}(\omega, \bar{\omega}) * \tilde{\mathbf{X}}\|^2 \right\} \quad (4.6.10)$$

subject to the following constraint:

$$\text{vec}^H(\mathbf{H}(\omega, \bar{\omega})) \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega}) = 1. \quad (4.6.11)$$

Minimization of the criterion (4.6.10) amounts to minimizing the energy of the data matrix \mathbf{X} when run backwards and forwards through the filter $\mathbf{H}(\omega, \bar{\omega})$. The constraint in (4.6.11) guarantees that the transfer function of the filter $\mathbf{H}(\omega, \bar{\omega})$ is equal to unity at the frequency $\omega, \bar{\omega}$ and hence that a sinusoid with frequency $\omega, \bar{\omega}$ passes through the filter undistorted.

Capon with FB-averaging has slightly worse resolution than Capon without FB-averaging [18], [19], [20].

APES: In a similar way, the APES spectrum estimate $\alpha(\omega, \bar{\omega})$ and the APES filter coefficient matrix $\mathbf{H}(\omega, \bar{\omega})$ are the minimizers of the following least-squares

(LS) criterion:

$$\min_{\mathbf{H}(\omega, \bar{\omega}), \alpha(\omega, \bar{\omega}), \beta(\omega, \bar{\omega})} \left\{ \left\| \text{vec}(\mathbf{H}(\omega, \bar{\omega}) * \mathbf{X}) - \alpha(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}(\omega, \bar{\omega}) \right\|^2 + \left\| \text{vec}(\mathbf{H}(\omega, \bar{\omega}) * \tilde{\mathbf{X}}) - \beta(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}(\omega, \bar{\omega}) \right\|^2 \right\} \quad (4.6.12)$$

subject to the (4.6.11) and the additional constraint:

$$\beta(\omega, \bar{\omega}) = \alpha^*(\omega, \bar{\omega}) e^{-i((N-1)\omega + (\bar{N}-1)\bar{\omega})} \quad (4.6.13)$$

where $\beta(\omega, \bar{\omega})$ can be interpreted as the amplitude of the sinusoid obtained by running the data *backwards* through the filter. Equation (4.6.13) expresses a constraint between the amplitude of the sinusoid fitted to the forward-filtered data and the amplitude of the sinusoid fitted to the backward-filtered data. It can be shown [12] that the constraint (4.6.13) is redundant, i.e., the minimizer of (4.6.10), subject to (4.6.11) only, necessarily satisfies (4.6.13).

FB averaging of the APES spectral estimate typically improves the spectral resolution [27] (see also Example 4.5 below).

4.7 Performance Analysis

The following discussion will provide some insight into the behavior of the different nonparametric estimators. For simplicity, we treat only the 1D versions of the estimators; with some additional notational effort the same results can be proved by the analysis of the estimators for 2D data. We start in Subsection 4.7.1 by proving that if M remains fixed while $N \rightarrow \infty$, all filterbank spectral estimators are statistically equivalent. This result was first established in [23] for the forward-only case and in [21] for the forward-backward case; a somewhat more general analysis is contained in [16]. Next, in Subsections 4.7.2 and 4.7.3 we go on to analyze the case of a *finite* number of samples N but a high signal-to-noise ratio. In this case, the properties of the different estimators are somewhat different from one another. The results in Subsection 4.7.2 are novel, whereas the results of Subsection 4.7.3 are due to [21].

4.7.1 Large-Sample Analysis

We assume that $x(n)$ is a signal comprising K complex sinusoids in noise, and hence that (4.2.2) holds:

$$x(n) = \sum_{k=1}^K \alpha_k e^{i\omega_k n} + w(n). \quad (4.7.1)$$

We also assume that $\{\alpha_k, \omega_k\}_{k=1}^K$ are deterministic quantities; hence the process $x(n)$ is cyclo-stationary [14]. Provided that

$$\frac{\Phi^{-1}(\omega)\mathbf{a}_M(\omega)}{\mathbf{a}_M^H(\omega)\Phi^{-1}(\omega)\mathbf{a}_M(\omega)} \rightarrow \mathbf{h}_0(\omega) \quad (4.7.2)$$

for some deterministic vector $\mathbf{h}_0(\omega)$ as $N \rightarrow \infty$ and M is fixed (which is a weak assumption), all filterbank estimators give asymptotically (in N or L) the same mean-square error (MSE) of the spectrum estimate for a fixed ω . Of course, if this MSE goes to zero as $N \rightarrow \infty$ then in particular the estimators give consistent estimates of the amplitude spectrum. We formalize this statement in the following theorem.

Theorem 4.1: *Assume that $x(n)$ is an outcome of (4.7.1), and let $\{\alpha_k\}$ be fixed (in other words, consider (4.7.1) conditioned on $\{\alpha_k\}$). Furthermore, let ω and M be fixed and assume that (4.7.2) holds, for some deterministic vector $\mathbf{h}_0(\omega)$, in a mean square sense as $L \rightarrow \infty$. Then*

$$E[|\hat{\alpha}(\omega) - \alpha(\omega)|^2] \rightarrow \frac{1}{L} \cdot P(\omega) \quad (4.7.3)$$

as $L \rightarrow \infty$, where $P(\omega)$ is the power spectrum of $w(n)$ at the frequency ω .

Proof: See Section 4.A.

We leave as an exercise to the reader to verify that (4.7.2) holds for all nonparametric estimators treated in this chapter, and hence that Theorem 4.1 is applicable to these estimators.

In principle, Theorem 4.1 states that all filterbank estimators are equally good for large N and therefore provides some justification for the somewhat heuristic arguments that were used to derive them. However, as we will see next, the *finite-sample* properties of the various estimators may differ substantially from what we would expect from Theorem 4.1. In particular, the resolution properties of the estimators are fairly different.

4.7.2 Leakage Analysis

For simplicity of the analysis, assume in this subsection that $x(n)$ is a single sinusoid with frequency ω_0 embedded in white noise $w(n)$ with variance σ^2 :

$$x(n) = e^{i\omega_0 n} + w(n). \quad (4.7.4)$$

Let

$$\mathbf{w} = [w(0) \ \cdots \ w(N-1)]^T \quad (4.7.5)$$

and consider first the DFT spectrum (4.3.15). When $\sigma^2 \rightarrow 0$,

$$\begin{aligned}\hat{\alpha}_{\text{DFT}}(\omega) &= \frac{1}{N} \mathbf{a}_N^H(\omega) (\mathbf{a}_N(\omega_0) + \mathbf{w}) \rightarrow \frac{1}{N} \sum_{n=0}^{N-1} e^{i\omega_0 n} e^{-i\omega n} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} e^{i(\omega_0 - \omega)n} = \begin{cases} 1 & \omega = \omega_0 \\ \frac{1}{N} \cdot \frac{1 - e^{i(\omega_0 - \omega)N}}{1 - e^{i(\omega_0 - \omega)}} & \omega \neq \omega_0 \end{cases}.\end{aligned}\quad (4.7.6)$$

Equation (4.7.6) illustrates the so-called *leakage* effect of the DFT: the presence of a sinusoid at a certain frequency ω_0 introduces a bias (even when the $\text{SNR} \rightarrow \infty$) in the spectrum estimate for *any* different frequency $\omega \neq \omega_0$. The DFT therefore gives inconsistent (in SNR) spectral estimates. The bias of the DFT spectrum can be thought of as a consequence of the fact that the set of vectors $\{\mathbf{a}_N(\omega)\}$ is not perfectly orthogonal to the vector $\mathbf{a}_N(\omega_0)$ when $\omega \neq \omega_0$. However, if $|\omega - \omega_0| > 2\pi/N$ the correlation between $\mathbf{a}_N(\omega)$ and $\mathbf{a}_N(\omega_0)$ is small in general, and hence the DFT is a useful estimator if the spectrum is known to consist of sinusoids that are sufficiently separated in frequency.

It is known that the leakage effect discussed in the above paragraph is significantly less for the other nonparametric estimators than for the DFT. The following discussion will help us understand why this is so. Consider, for the sake of illustration, the Capon estimator (4.3.19). For high SNR, $M > 1$, and a large but fixed number of snapshots L , we have that:

$$\begin{aligned}\sigma^2 \hat{\mathbf{R}}^{-1} &\approx \sigma^2 \mathbf{R}^{-1} = \left(\mathbf{I} + \frac{1}{\sigma^2} \mathbf{a}_M(\omega_0) \mathbf{a}_M^H(\omega_0) \right)^{-1} \\ &= \mathbf{I} - \frac{1}{\sigma^2 + \mathbf{a}_M^H(\omega_0) \mathbf{a}_M(\omega_0)} \mathbf{a}_M(\omega_0) \mathbf{a}_M^H(\omega_0) \\ &\approx \mathbf{I} - \frac{\mathbf{a}_M(\omega_0) \mathbf{a}_M^H(\omega_0)}{\mathbf{a}_M^H(\omega_0) \mathbf{a}_M(\omega_0)} = \Pi_{\mathbf{a}_M(\omega_0)}^\perp\end{aligned}\quad (4.7.7)$$

where $\Pi_{\mathbf{X}} = \mathbf{X}(\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H$ and $\Pi_{\mathbf{X}}^\perp = \mathbf{I} - \Pi_{\mathbf{X}}$ stand for the orthogonal projection onto the column space of \mathbf{X} and its complement, respectively, and where in the last equality we have used the matrix inversion lemma [17], [5]. It follows that

$$\hat{\alpha}_{\text{Capon}}(\omega) = \frac{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{a}_M(\omega)} \approx \frac{\mathbf{a}_M^H(\omega) \Pi_{\mathbf{a}_M(\omega_0)}^\perp \mathbf{g}(\omega)}{\mathbf{a}_M^H(\omega) \Pi_{\mathbf{a}_M(\omega_0)}^\perp \mathbf{a}_M(\omega)}.\quad (4.7.8)$$

The usefulness of the approximation in (4.7.8) is somewhat limited since the denominator goes to zero as $\omega \rightarrow \omega_0$. However, for $|\omega - \omega_0| \neq 0$ and $M > 1$, the denominator of (4.7.8) is nonzero. Moreover, since

$$\mathbf{g}(\omega) \rightarrow \frac{1}{L} \mathbf{a}_M(\omega_0) \mathbf{a}_L^T(\omega_0) \mathbf{a}_L^*(\omega)\quad (4.7.9)$$

as $\sigma^2 \rightarrow 0$ (see (4.3.5), (4.3.7) and (4.3.11)), it follows that

$$\Pi_{\mathbf{a}_M(\omega_0)}^\perp \mathbf{g}(\omega) \rightarrow \mathbf{0}\quad (4.7.10)$$

as $\text{SNR} \rightarrow \infty$ and hence the numerator (4.7.8) goes to zero under the same assumption. This analysis indicates that the Capon spectrum is close to zero for all $\omega \neq \omega_0$ and high SNR. Consequently Capon does not suffer from the same serious leakage effects as the DFT does. Another way to understand this observation is that for the estimation of the spectrum for $\omega \neq \omega_0$, the covariance matrix \mathbf{R} will model the contribution to $x(n)$ from other frequencies than ω (in particular from the sinusoid at ω_0), and this contribution will be suppressed by the weighting in the WLS criterion (4.3.9).

Note that in the case of $M = 1$, Capon reduces to the DFT. In this case, the above analysis is not applicable (if $M = 1$, the denominator of (4.7.8) is zero for all ω).

Another intuitively appealing explanation of the absence of leakage for Capon is offered by the filterbank interpretation in Section 4.4. It turns out that for the estimation of $\alpha(\omega)$, the filter associated with $\mathbf{h}(\omega)$ has (by construction) a unit gain at the frequency ω of interest, but typically a gain close to zero at all other frequencies where significant power is present. In other words, the filter tends to annihilate spectral components that may potentially interfere with the particular one under study. We will exemplify this fact in Example 4.2 in Section 4.9 below.

4.7.3 Bias Analysis

The following theorem (from [21]) is interesting since it establishes an analytical explanation of the empirically observed fact that APES is less biased than Capon.

Theorem 4.2: *Suppose that the third order moments of $w(n)$ are equal to zero. Then, to within a second-order approximation, the biases of the Capon and APES estimates obey the following relations.*

(a) *Capon is biased downwards, i.e.,*

$$E[\hat{\alpha}_{\text{Capon}}(\omega)] \lesssim \alpha(\omega). \quad (4.7.11)$$

(b) *The bias of FB-Capon is half that of F-Capon, that is,*

$$E[\hat{\alpha}_{\text{FB-Capon}}(\omega)] - \alpha(\omega) \approx \frac{E[\hat{\alpha}_{\text{F-Capon}}(\omega)] - \alpha(\omega)}{2}. \quad (4.7.12)$$

(c) *APES is unbiased, i.e.,*

$$E[\hat{\alpha}_{\text{APES}}(\omega)] \approx \alpha(\omega). \quad (4.7.13)$$

Proof: See Section 4.B.

4.8 Fast Implementations of Capon and APES

A direct implementation of the filterbank estimators by computing (4.3.10) in a direct way for many different ω is computationally demanding, and several papers in the literature have dealt with the problem of implementing Capon and APES efficiently. For instance, in [19], [20] a fast (but approximate) implementation of Capon was obtained by replacing the outer-product sample covariance matrix with a structured block-Toeplitz matrix obtained by fitting an autoregressive model to the data.⁴ The implementation of APES was discussed in [30], and to some extent, in [21]. Although the technique in [30] improves over the direct way of implementing APES suggested in [21], its computational complexity is still very high and typically it must be combined with suboptimal splitting techniques such as that in [31], or with other approximations.

The fastest available technique for the computation of the Capon and APES spectra is that of [32]. For APES, the techniques in [32] are several orders of magnitude faster than those in [30], [21] whereas for Capon the improvement over the approximate techniques in [19], [28] is not fully as large. The implementations in [32] also reduce the memory demands for the computation significantly; using the algorithms therein quite large 2D spectra can be computed on a standard personal computer. This chapter summarizes the main results from [32]. We treat only the 2D case with FB averaging; the 1D case follows as a special case and the variants without FB averaging can be obtained by omitting certain terms from the derived expressions.

4.8.1 Some Results on Fast Computation

The starting point for the derivation of fast implementations is the following result.

Lemma 4.1: *Define the following trigonometric polynomials:*

$$\begin{aligned}
 p_{11}(\omega, \bar{\omega}) &\triangleq \mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{R}}^{-1} \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega}) \\
 p_{12}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{R}}^{-1} \mathbf{Y} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega}) \\
 p_{13}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \mathbf{a}_{M, \bar{M}}^H(\omega, \bar{\omega}) \hat{\mathbf{R}}^{-1} \tilde{\mathbf{Y}} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega}) \\
 p_{22}(\omega, \bar{\omega}) &\triangleq \frac{1}{L^2 \bar{L}^2} \mathbf{a}_{L, \bar{L}}^H(-\omega, -\bar{\omega}) \mathbf{Y}^H \hat{\mathbf{R}}^{-1} \mathbf{Y} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega}) \\
 p_{23}(\omega, \bar{\omega}) &\triangleq \frac{1}{L^2 \bar{L}^2} \mathbf{a}_{L, \bar{L}}^H(-\omega, -\bar{\omega}) \mathbf{Y}^H \hat{\mathbf{R}}^{-1} \tilde{\mathbf{Y}} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega})
 \end{aligned} \tag{4.8.1}$$

and the trigonometric polynomial matrix

$$\Sigma(\omega, \bar{\omega}) \triangleq \frac{1}{2} \begin{bmatrix} p_{22}(\omega, \bar{\omega}) - 2 & p_{23}(\omega, \bar{\omega}) \\ p_{23}^*(\omega, \bar{\omega}) & p_{22}(\omega, \bar{\omega}) - 2 \end{bmatrix}. \tag{4.8.2}$$

⁴A similar idea was used in [28] to obtain a fast implementation of the PSC estimator (which is *not* equivalent to the Capon method described in this chapter; cf. the remark at the end of Section 4.3.3); in the 1D case the technique of [28] is equivalent to that in [29].

Then it holds that

$$p_{13}^*(\omega, \bar{\omega}) = p_{12}^*(\omega, \bar{\omega}) e^{-i((N-1)\omega + (\bar{N}-1)\bar{\omega})}. \quad (4.8.3)$$

Furthermore, the FB-averaged Capon and APES spectra can be expressed as

$$\hat{\alpha}_{\text{Capon}}(\omega, \bar{\omega}) = \frac{p_{12}(\omega, \bar{\omega})}{p_{11}(\omega, \bar{\omega})} \quad (4.8.4)$$

and

$$\hat{\alpha}_{\text{APES}}(\omega, \bar{\omega}) = \frac{p_{12}(\omega, \bar{\omega}) - \frac{1}{2} [p_{12}(\omega, \bar{\omega}) \quad p_{13}(\omega, \bar{\omega})] \boldsymbol{\Sigma}^{-1}(\omega, \bar{\omega}) \begin{bmatrix} p_{22}(\omega, \bar{\omega}) \\ p_{23}^*(\omega, \bar{\omega}) \end{bmatrix}}{p_{11}(\omega, \bar{\omega}) - \frac{1}{2} [p_{12}(\omega, \bar{\omega}) \quad p_{13}(\omega, \bar{\omega})] \boldsymbol{\Sigma}^{-1}(\omega, \bar{\omega}) \begin{bmatrix} p_{12}^*(\omega, \bar{\omega}) \\ p_{13}^*(\omega, \bar{\omega}) \end{bmatrix}}. \quad (4.8.5)$$

Proof: Equation (4.8.3) is shown in [21]. The expressions in (4.8.4) follow immediately from (4.6.8) and (4.8.1). The proof of equation (4.8.5) consists of applying the matrix inversion lemma to (4.6.9).

Let $\hat{\mathbf{R}}^{-1/2}$ denote the inverse of the Cholesky factor of $\hat{\mathbf{R}}$, i.e., the (unique) upper triangular matrix with real and positive diagonal elements such that

$$\hat{\mathbf{R}}^{-1/2} (\hat{\mathbf{R}}^{-1/2})^H = \hat{\mathbf{R}}^{-1}. \quad (4.8.6)$$

Furthermore, let $\{\mathbf{r}_m\}_{m=1}^{M\bar{M}}$ be the columns of $\hat{\mathbf{R}}^{-1/2}$. Then the polynomials in Lemma 4.1 can be written as:

$$p_{ij}(\omega, \bar{\omega}) = \sum_{m=1}^{M\bar{M}} p_i^{(m)*}(\omega, \bar{\omega}) p_j^{(m)}(\omega, \bar{\omega}) = \sum_{m=1}^{M\bar{M}} p_{ij}^{(m)}(\omega, \bar{\omega}) \quad (4.8.7)$$

where

$$\begin{aligned} p_1^{(m)}(\omega, \bar{\omega}) &\triangleq \mathbf{r}_m^H \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega}) \\ p_2^{(m)}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \mathbf{r}_m^H \mathbf{Y} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega}) \\ p_3^{(m)}(\omega, \bar{\omega}) &\triangleq \frac{1}{L\bar{L}} \mathbf{r}_m^H \tilde{\mathbf{Y}} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega}) \end{aligned} \quad (4.8.8)$$

and

$$p_{ij}^{(m)}(\omega, \bar{\omega}) \triangleq p_i^{(m)*}(\omega, \bar{\omega}) p_j^{(m)}(\omega, \bar{\omega}). \quad (4.8.9)$$

The technique proposed in [30] for the computation of 1D and 2D APES and Capon evaluates the polynomials in Lemma 4.1 by using the fact that once $\hat{\mathbf{R}}^{-1/2}$, $\hat{\mathbf{R}}^{-1/2} \mathbf{Y}$ and $\hat{\mathbf{R}}^{-1/2} \tilde{\mathbf{Y}}$ were computed, the trigonometric polynomial vectors $\hat{\mathbf{R}}^{-1/2} \mathbf{a}_{M, \bar{M}}(\omega, \bar{\omega})$, $\hat{\mathbf{R}}^{-1/2} \mathbf{Y} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega})$ and $\hat{\mathbf{R}}^{-1/2} \tilde{\mathbf{Y}} \mathbf{a}_{L, \bar{L}}(-\omega, -\bar{\omega})$ can be evaluated using $M\bar{M}$ 2D FFTs of size $K \times \bar{K}$. In [20], the same technique was used for

the evaluation of the Capon spectrum, with the exception that the Cholesky factor $\hat{\mathbf{R}}^{-1/2}$ is approximated with a structured matrix obtained by fitting an autoregressive model to the data. This approximation technique provides efficient means for computing $\hat{\mathbf{R}}^{-1/2}$, which was sometimes claimed to be a bottleneck in the computation of the spectrum (at least for the 1D case). However, it turned out that this technique is not applicable to the computation of the APES spectrum.

The key observation that leads to our fast implementation of Capon and APES is the following. Instead of computing the values of the $M\bar{M}$ polynomials in (4.8.8) at $K \times \bar{K}$ points, we compute the coefficients of the polynomials $p_{ij}(\omega, \bar{\omega})$ in (4.8.7) and upon completion of this computation, each polynomial is evaluated at $K \times \bar{K}$ points using a *single* 2D FFT. The computation of the polynomial coefficients is performed by accumulating the contributions from $p_{ij}^{(m)}(\omega, \bar{\omega})$ in the sum in (4.8.7).

To exploit this idea optimally, it must be combined with some relevant results on fast computation involving matrices and trigonometric polynomials.

Fast Matrix-Vector Multiplication

The matrices \mathbf{Y} and $\tilde{\mathbf{Y}}$ have a structure that enables products of the type $\mathbf{u}^H \mathbf{Y}$, $\mathbf{u}^H \tilde{\mathbf{Y}}$, $\mathbf{Y} \mathbf{u}$ and $\tilde{\mathbf{Y}} \mathbf{u}$ (for arbitrary vectors \mathbf{u} of compatible dimension) to be evaluated in $O(N\bar{N} \log(N\bar{N}))$ floating point operations. The procedure for doing so is reminiscent of that for fast convolution, and is outlined in Section 4.C. Note that a direct evaluation of the vector-matrix product $\mathbf{u}^H \mathbf{Y}$ would require $O(M\bar{M}N\bar{N})$ operations and hence the technique in Section 4.C is considerably faster.

An important consequence of this result is that $\hat{\mathbf{R}}$ can be computed in $O(M\bar{M}N\bar{N} \log(N\bar{N}))$ operations. To see this, note that row k of $\hat{\mathbf{R}}$ equals $1/(L\bar{L}) \mathbf{u}_k^H \mathbf{Y}^H$ where \mathbf{u}_k^H is the k th row of \mathbf{Y} , and apply the fast matrix-vector multiplication method $M\bar{M}$ times (once for each row). This should be compared with direct evaluation of (4.6.3) and (4.6.5) which requires $O(M^2\bar{M}^2N\bar{N})$ operations.

Fast Multiplication of Trigonometric Polynomials

Consider two 2-D trigonometric polynomials of the form:

$$\begin{aligned} p(\omega, \bar{\omega}) &= p_{n_p-1, \bar{n}_p-1} e^{i((n_p-1)\omega + (\bar{n}_p-1)\bar{\omega})} + \dots \\ &\quad + p_{n_p-1, 0} e^{i(n_p-1)\omega} + \dots + p_{0, \bar{n}_p-1} e^{i(\bar{n}_p-1)\bar{\omega}} + \dots + p_{0, 0} \end{aligned} \quad (4.8.10)$$

and

$$\begin{aligned} q(\omega, \bar{\omega}) &= q_{n_q-1, \bar{n}_q-1} e^{i((n_q-1)\omega + (\bar{n}_q-1)\bar{\omega})} + \dots \\ &\quad + q_{n_q-1, 0} e^{i(n_q-1)\omega} + \dots + q_{0, \bar{n}_q-1} e^{i(\bar{n}_q-1)\bar{\omega}} + \dots + q_{0, 0} \end{aligned} \quad (4.8.11)$$

Then the $(n_p + n_q - 1)(\bar{n}_p + \bar{n}_q - 1)$ coefficients of the following polynomial:

$$\begin{aligned} r(\omega, \bar{\omega}) = p^*(\omega, \bar{\omega})q(\omega, \bar{\omega}) &= r_{n_q-1, \bar{n}_q-1} e^{i((n_q-1)\omega + (\bar{n}_q-1)\bar{\omega})} + \dots \\ &+ r_{0,0} + \dots + r_{-n_p+1, -\bar{n}_p+1} e^{i((-n_p+1)\omega + (-\bar{n}_p+1)\bar{\omega})} \end{aligned} \quad (4.8.12)$$

as well as the coefficients of the polynomials

$$\begin{aligned} \tilde{r}(\omega, \bar{\omega}) &= p^*(\omega, \bar{\omega})q(-\omega, -\bar{\omega}) \\ \hat{r}(\omega, \bar{\omega}) &= p^*(-\omega, -\bar{\omega})q(-\omega, -\bar{\omega}) \end{aligned} \quad (4.8.13)$$

can be computed in $O\left((n_p + n_q)(\bar{n}_p + \bar{n}_q) \log((n_p + n_q)(\bar{n}_p + \bar{n}_q))\right)$ operations. This result, which we prove in Section 4.D, is based on an isomorphism between multiplication of polynomials and a standard convolution (which can be performed using the fast Fourier transform (FFT)).

Note that a direct computation of the polynomial coefficients would require $O((n_p + n_q)^2(\bar{n}_p + \bar{n}_q)^2)$ operations.

Fast Cholesky Factorization of $\hat{\mathbf{R}}^{-1}$

The matrix $\hat{\mathbf{R}}$ is rich in structure, and this structure can be exploited in order to compute its Cholesky factorization and inverse. Using a certain *displacement factorization* of $\hat{\mathbf{R}}$ presented in [32], it is possible to apply the fast generalized Schur algorithm of [33], [34]. The resulting fast algorithm for fast Cholesky factorization of $\hat{\mathbf{R}}^{-1}$ is given in Section 4.E.

By using the technique of Section 4.E, the columns $\{\mathbf{r}_k\}_{k=1}^{M\bar{M}}$ of $\hat{\mathbf{R}}^{-1/2}$ can be computed *from the data* in $O(MN\bar{N} \log(N\bar{N}) + M^2\bar{M}^2N)$ floating point operations. This operation count should be compared to a direct computation of $\hat{\mathbf{R}}$ via (4.6.3) and (4.6.5) followed by a standard Cholesky factorization and inversion [35] which together require $O(M^2\bar{M}^2N\bar{N})$ operations. It turns out that in the 1D case, the method of Section 4.E usually is significantly faster than inversion and classical Cholesky factorization. This is true also for the 2D case, provided that M, \bar{M} are relatively large (which in effect is the case of interest in practice), although for small values of M, \bar{M} conventional Cholesky factorization and inversion can be somewhat faster.

Under the assumption, which is valid in our application, that each \mathbf{r}_k is used directly after its computation, running the algorithm requires only $O(M\bar{M}N)$ bits of memory. Note that the storage of the whole Cholesky factor would require $O(M^2\bar{M}^2)$ bits of memory, which is typically much more than the storage requirement of our algorithm. The low memory requirement (especially in the 1D case) of the algorithm may be of value for both possible hardware implementations and off-line data analysis applications.

Finally, we stress that the method of Section 4.E gives the inverse of the exact Cholesky factor, which should not be confused with the approximations in [28],

[20] (the latter are also relatively computationally efficient but they operate on a structured Toeplitz approximation of $\hat{\mathbf{R}}$).

4.8.2 Fast Implementation

1. Decide, based on the values of M, \bar{M}, N, \bar{N} and the above operation counts, whether to use classical Cholesky factorization and inversion or the fast technique of Section 4.E to compute $\hat{\mathbf{R}}^{-1/2}$. If the fast factorization method is chosen, compute the displacement factorizations $\{\tilde{\mathbf{G}}, \Psi, M\}$ introduced in Section 4.E. Otherwise, first compute $\hat{\mathbf{R}}$ as indicated in the previous subsection and thereafter compute $\hat{\mathbf{R}}^{-1/2}$ by a direct method [35].
2. Perform the following steps for $m = 1, \dots, M\bar{M}$:
 - (a) Obtain the m th column \mathbf{r}_m of $\hat{\mathbf{R}}^{-1/2}$:
 - If the classical Cholesky factorization is used, \mathbf{r}_m is already precomputed in Step 1 above.
 - If the fast Cholesky factorization is used, perform one iteration (i.e. the Steps 1-3) of the generalized Schur algorithm described in Section 4.E.
 - (b) Compute $\mathbf{r}_m^H \mathbf{Y}$ and $\mathbf{r}_m^H \tilde{\mathbf{Y}}$ by using the technique in Section 4.C.
 - (c) Compute the coefficients of the polynomials $p_1^{(m)}(\omega, \bar{\omega})$, $p_2^{(m)}(\omega, \bar{\omega})$, $p_3^{(m)}(\omega, \bar{\omega})$ in (4.8.8) by using the results of step (a)-(b) above.
 - (d) Compute the coefficients of the polynomials $p_{ij}^{(m)}(\omega, \bar{\omega})$ in (4.8.9) by using the technique in Section 4.D. These coefficients are summed up to obtain the coefficients of the polynomials $p_{ij}(\omega, \bar{\omega})$ in (4.8.7).
3. Evaluate the polynomials $p_{ij}(\omega, \bar{\omega})$ at $K \times \bar{K}$ points by using a single 2D FFT (per polynomial).
4. Compute the Capon or APES spectrum according to (4.8.4) or (4.8.5).

4.8.3 Discussion

Numerical examples presented in [32] (but omitted here for brevity) demonstrate that the implementations described above improve over the techniques in, for instance, [30], [20] by several orders of magnitude. In fact, the difference in computational complexity between these implementations increases with increasing data sample lengths, and with increasing K/N and \bar{K}/\bar{N} . Note in particular that the implementation outlined above is practically insensitive to the density of the frequency grid (K, \bar{K}) , unlike those in [30], [28], [20].

The fast algorithms operate at maximum efficiency if N, \bar{N} and M, \bar{M} are chosen such that $N, \bar{N}, 2M - 1, 2\bar{M} - 1, 2L - 1$ as well as $2\bar{L} - 1$ approach from below a power of two. The reason for this is that the FFTs used in the above fast methods

are then exploited optimally (it is possible to overcome this restriction by using non-power of two variants of the FFT). Furthermore, K and \bar{K} should be powers of two. Finally, if the fast Cholesky factorization is used, the orientation of the data matrix should be chosen such that $L \leq \bar{L}$. Further remarks and details that can speed up the implementation are given in [36].

4.9 Performance Examples and Applications of Capon and APES

In this section, we present numerical examples showing the performance of the APES and Capon algorithms in complex spectral estimation exercises as well as in applications to radar range signature estimation and synthetic aperture radar (SAR) imaging.

4.9.1 1D Examples

Example 4.1: DFT, Capon and APES applied to a 1D time-series.

In our first example (see also [11]) we compare the performances of Capon and APES with those of the DFT, windowed DFT and Welch methods for the estimation of a 1D line spectrum. The data sequence is $N = 128$ samples long and consists of 4 dominant spectral lines and many small closely spaced lines embedded in zero-mean white Gaussian noise with unit variance. The four dominant spectral lines are located at the frequencies 0.04, 0.05, 0.08, and 0.09 Hz, where Hz means cycles-per-sample. The estimation of these spectral components is our primary goal. Although we are interested in estimating the *complex* spectrum, which is important in range signature estimation and SAR imaging applications, such a spectrum is difficult to visualize effectively and therefore we show only the modulus of the complex spectral estimates.

The modulus of the true complex spectrum is shown in Figure 4.1(a). Figure 4.1(b) shows the modulus of the spectral estimate obtained via the DFT. We note that the use of the DFT method results in high sidelobes which, for instance, makes it hard to distinguish the peak caused by the 4th line in Figure 4.1(a) from the sidelobe peaks. Note also that the peak heights given by the DFT method are rather inaccurate estimates of the true ones shown in Figure 4.1(a). Figure 4.1(c) shows the modulus of the spectral estimate obtained by using a DFT with a Kaiser window and shape parameter 4. We can see that this windowed DFT method reduces the sidelobes of the spectral peaks. Yet, at the same time, the windowed DFT widens the spectral peaks and as a result, the third and fourth peaks in Figure 4.1(c) are smeared together. Figure 4.1(d) shows the modulus of the spectral estimate obtained by using the unwrapped Welch method (or averaged DFT) with $M = 64$. We note that the Welch method performs similarly to the windowed FFT method. Finally, Figure 4.1(e) shows the modulus of the spectral estimate obtained by using the Capon method with $M = 64$ and Figure 4.1(f) shows the modulus of the spectral estimate obtained by using the APES algorithm with $M = 64$.

Although the spectral peaks given by Capon are very narrow, the corresponding estimated peak heights are less accurate than those obtained via the windowed DFT or Welch methods. The sidelobes in the APES spectral estimate are comparable with those in the Capon spectrum method, but the spectral peaks given by APES are somewhat wider than those given by Capon. Note also that the amplitude estimates *at the true locations of the sinusoids* obtained via APES are more

accurate than those obtained from Capon or DFT. In fact, in this example the APES algorithm gives the most accurate peak height estimates among all methods considered. The Capon estimates are slightly biased downwards (cf. Theorem 4.2). Finally, note that except for the DFT method, all other methods considered herein tend to suppress the small closely spaced spectral lines. ■

We shall remark on the fact that, although this is not apparent from Example 4.1, it has been observed in other studies that compared to DFT, its averaged relatives as well as APES, the Capon estimate appears to yield superior precision of the peak locations.

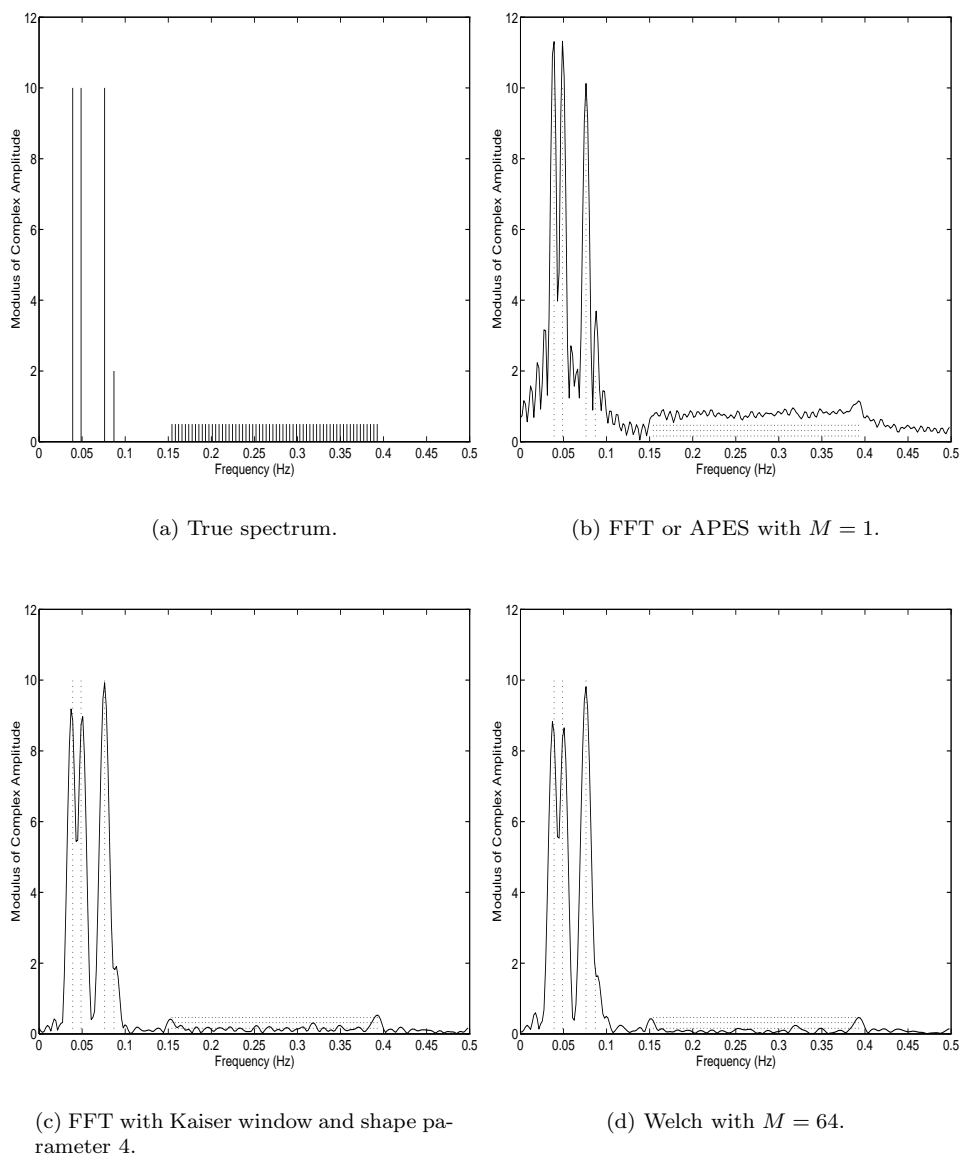


Figure 4.1. (continued on the next page)

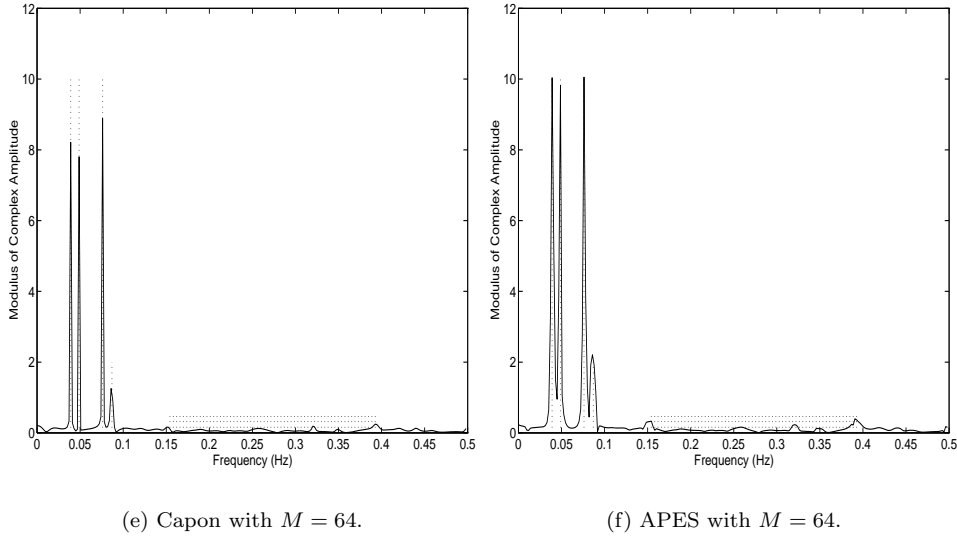


Figure 4.1. (cont.) Results for Example 4.1.

Example 4.2: The transfer function of the adaptive FIR filter.

In this example (see also [11]) we study the transfer function associated with the adaptive FIR filter $\mathbf{h}(\omega)$ for the estimation tasks in Example 4.1. Figure 4.2(a) shows the true spectrum (the same as in Figure 4.1(a)). Figures 4.2(b)–4.2(d) show the modulus of the frequency responses $|\mathbf{h}^H(\omega_0)\mathbf{a}(\omega)|$ of the FIR filters associated with the Welch method as well as Capon and APES. We show the transfer function as a function of $f = \omega/(2\pi)$, where $\omega_0 = 2\pi f_0$ denotes the frequency of interest. In Figure 4.2 we choose $f_0 = 0.04$ (corresponding to the frequency of the first spectral line in Figure 4.2(a)).

Note that all filters (by construction) satisfy the constraint $|\mathbf{h}^H(\omega_0)\mathbf{a}(\omega_0)| = 1$. We see from Figure 4.2(b) that $\mathbf{h}_{\text{Welch}}(\omega)$ does not annihilate the contributions from other spectral components. This is so since the Welch method is not *adaptive*, and the interfering spectral lines contribute to the error in $\hat{a}(\omega_0)$ by entering the filter output through both the mainlobe and the sidelobes of $\mathbf{h}_{\text{Welch}}(\omega)$. From Figures 4.2(c) and 4.2(d) we note that both $\mathbf{h}_{\text{Capon}}(\omega)$ and $\mathbf{h}_{\text{APES}}(\omega)$ nullify (to a large degree) the contributions from other spectral lines. This is a consequence of the adaptive nature of these methods. Yet the frequency responses of $\mathbf{h}_{\text{Capon}}(\omega)$ at frequencies other than ω_0 tend to be much higher than those of $\mathbf{h}_{\text{APES}}(\omega)$. These high sidelobes seem to cause the $\hat{a}(\omega)$ obtained with Capon to be less accurate than that obtained with APES, as may be seen from Figure 4.1. ■

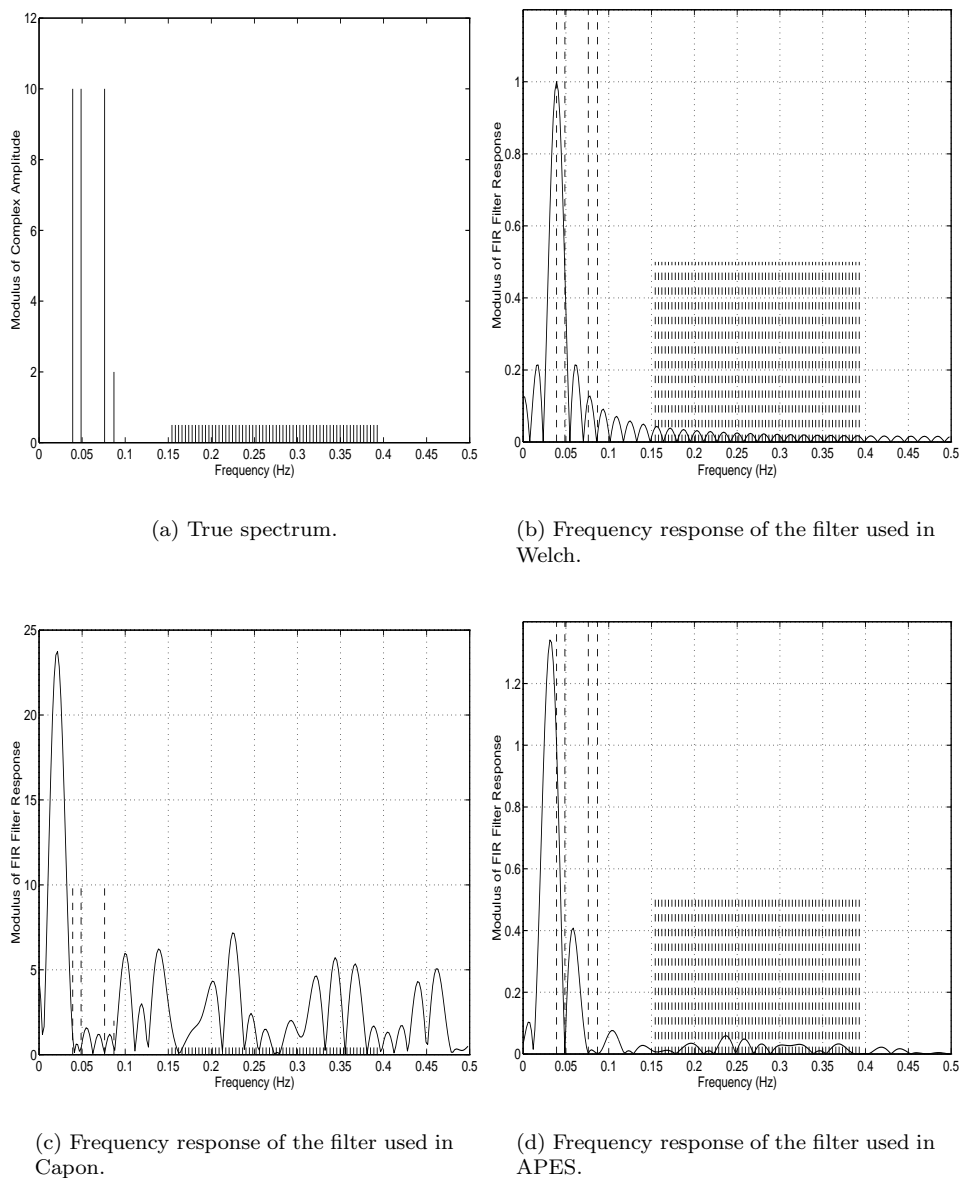
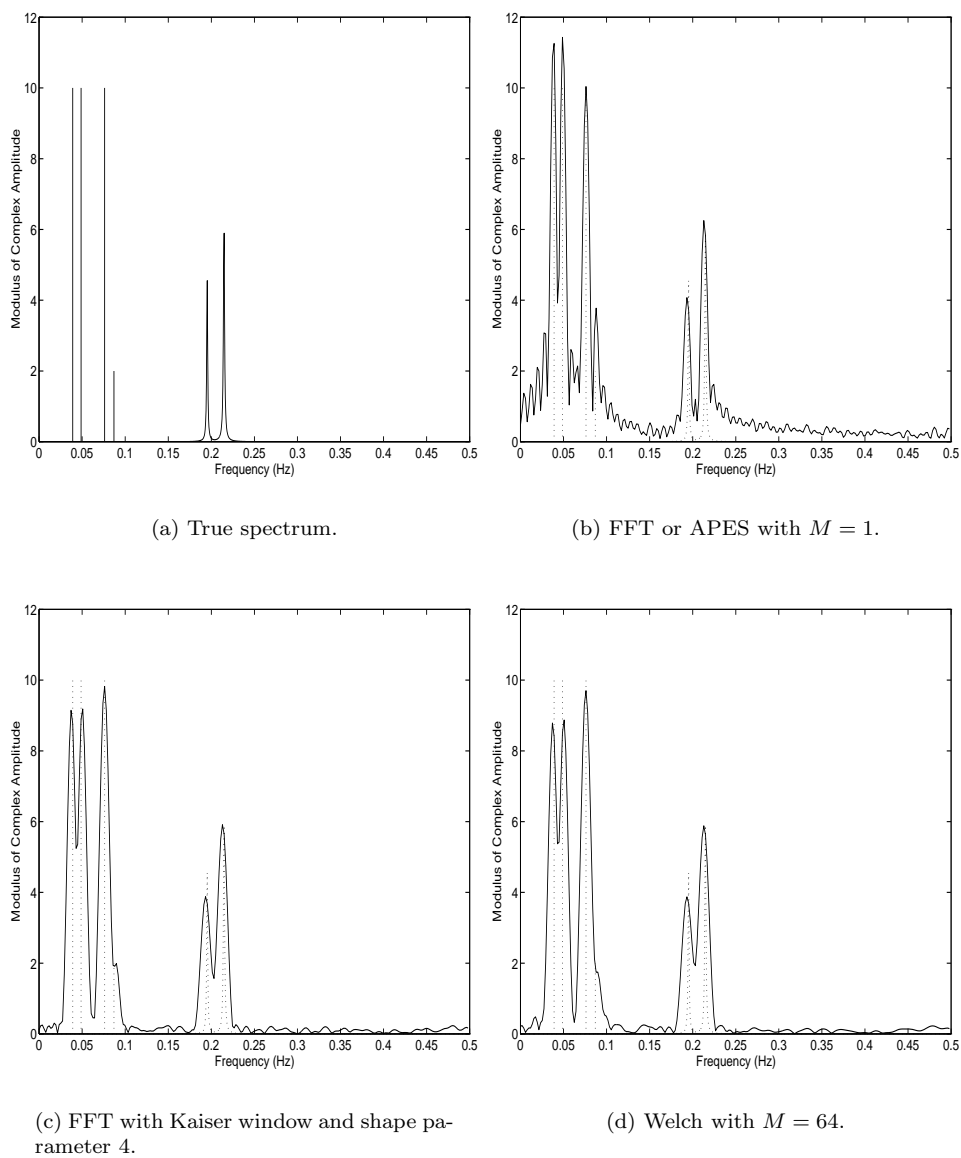


Figure 4.2. Results for Example 4.2. The figure shows the modulus of the frequency responses of the FIR filters in Example 4.1, when the first spectral line is the one of interest. Note that the scales are different in the different subfigures.

Example 4.3: Mixed-spectrum estimation (1D case).

We next consider the task of estimating a *mixed* spectrum that consists of both continuous components and a line spectrum (see also [11]). The data sequence under consideration has $N = 128$ samples and is corrupted by additive zero-mean white Gaussian noise with variance 1. The modulus of the true complex spectrum is shown in Figure 4.3(a). The two continuous parts of the spectrum in Figure 4.3(a) are caused by two damped sinusoids with damping coefficients 0.995 and 0.996. Figures 4.3(b)–4.3(f) show the spectral estimates obtained with the FFT, windowed FFT, Welch, Capon, and APES methods. The APES algorithm again gives the best spectral estimate. Figure 4.4 is very similar to Figure 4.3 except that in Figure 4.4 the continuous part of the spectrum is a rectangular pulse, which corresponds to a sinc-function in the time-domain.

From Figures 4.3 and 4.4 we see that the continuous spectral estimate obtained with the APES algorithm is similar to the ones obtained with the windowed FFT and Welch methods. The line spectral estimate obtained with the APES algorithm, however, is much better than those obtained with the windowed FFT and the Welch methods. The Capon method performs rather poorly on this data set. The APES algorithm again gives the best overall spectral estimate. ■

**Figure 4.3.** (continued on the next page)

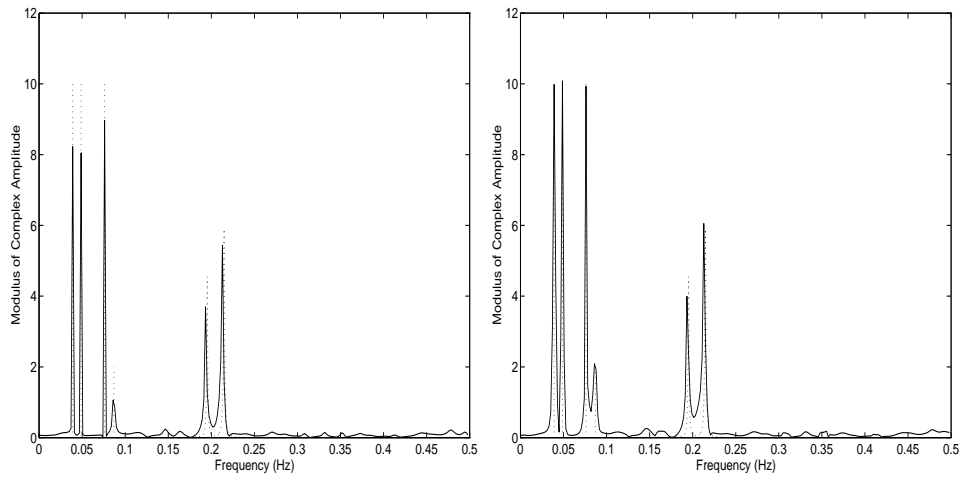
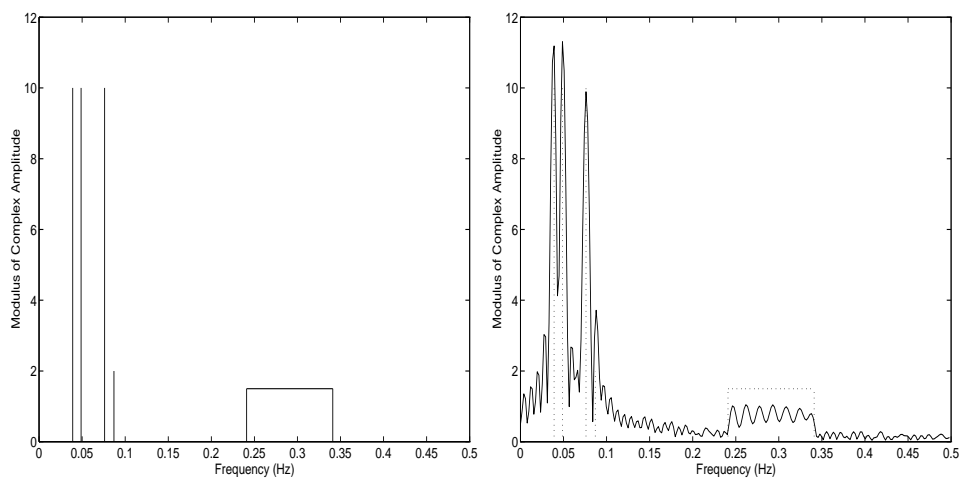
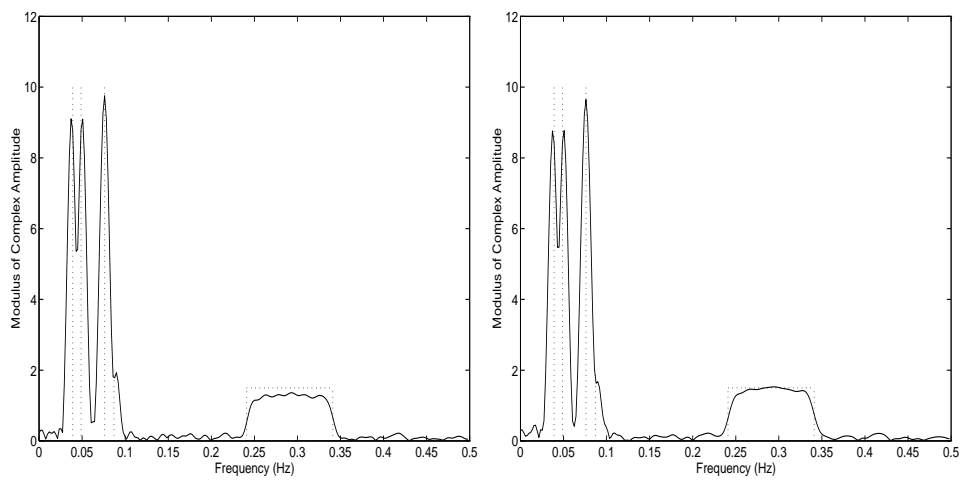
(e) Capon with $M = 64$.(f) APES with $M = 64$.

Figure 4.3. (cont.) Results from Example 4.3. In this example, the continuous spectral component corresponds to a damped sinusoid.



(a) True spectrum.

(b) FFT or APES with $M = 1$.

(c) FFT with Kaiser window and shape parameter 4.

(d) Welch with $M = 64$.**Figure 4.4.** (continued on the next page)

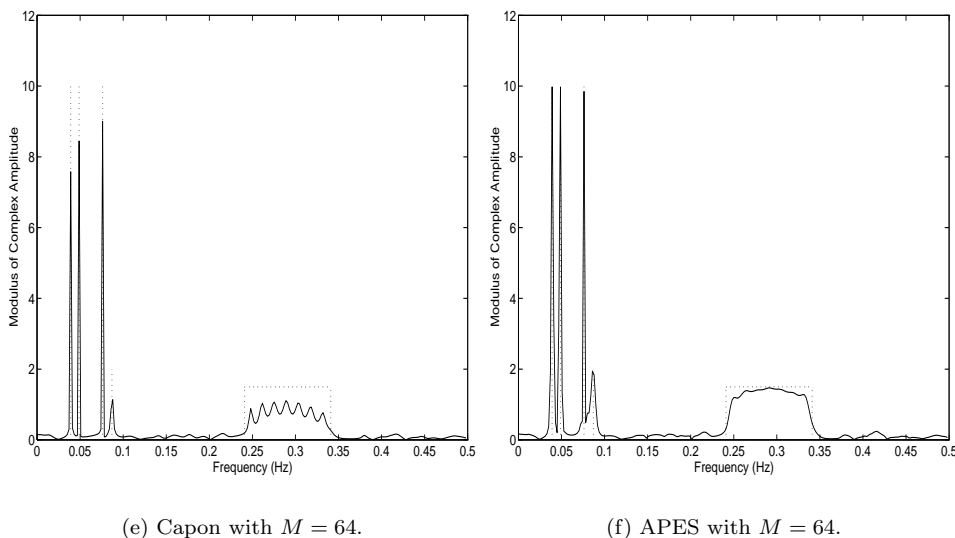


Figure 4.4. (cont.) Results from Example 4.3. In this example, the continuous spectral component corresponds to a damped sinc-function.

In the next example we will apply the adaptive FIR filtering methods to the problem of radar target range signature estimation. Before doing so, we briefly describe the target range signature estimation problem. In principle, the range resolution of a radar is determined by the radar bandwidth [37]. To achieve a high resolution in range, the radar must transmit a wideband pulse, which is often chosen to be a linearly frequency modulated (FM) chirp pulse. Such a chirp pulse can be written as (after appropriate normalization):

$$s(t) = \exp\left(-i(2\pi f_0 t + \gamma t^2)\right) \quad (4.9.1)$$

for $|t| \leq T_0/2$ where f_0 is the carrier frequency, 2γ is the FM rate and T_0 is the length of the time interval under consideration. The noise-free chirp pulse returned by a scatterer of a radar target can be written as

$$r(t) = \alpha \exp\left(-i(2\pi f_0(t - \tau) + \gamma(t - \tau)^2)\right), \quad (4.9.2)$$

where τ denotes the round-trip time delay between the scatterer and the radar and α stands for the radar cross section (RCS) of the scatterer. Note that the time delay τ and the range d of the scatterer are related as follows:

$$d = \frac{\tau c}{2}, \quad (4.9.3)$$

where c denotes the propagation speed of the transmitted electromagnetic wave. Mixing $r(t)$ with the reference chirp signal $s^*(t)$ at the receiver gives the following demodulated signal:

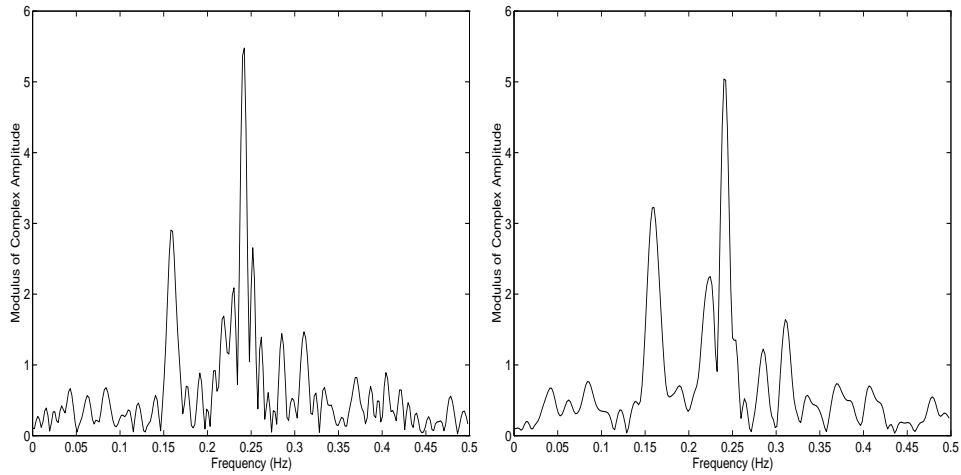
$$r_1(t) = \alpha \exp\left(i(2\pi f_0\tau - \gamma\tau^2)\right) \cdot \exp(i2\gamma\tau t) \quad (4.9.4)$$

which is a sinusoidal signal with frequency $2\gamma\tau$. Let τ_{\max} and τ_{\min} correspond to the maximum and the minimum values of the round-trip time delays between the scatterers of a radar target and the radar, respectively, and assume that $(\tau_{\max} - \tau_{\min}) \ll T_0$. Then for $-T_0/2 + \tau_{\max} \leq t \leq T_0/2 + \tau_{\min}$, the scatterers of the radar target at different ranges correspond to different frequencies of the demodulated signal. By definition, the target range signature is given by the RCS's of the target scatterers as a function of the range. Thus the problem of estimating the target range signature is equivalent to that of estimating the complex spectrum of the demodulated signal. This estimated complex target range signature can be useful for target recognition.

Note that for a radar with a very large bandwidth, α may change slightly as a function of time since the RCS of a scatterer changes slightly as the radar frequency changes. In this case, the peaks in the estimated range signature no longer correspond to the time-varying RCS of the scatterers and hence the estimated range signature cannot be used to accurately determine the time-varying RCS of the scatterers. Nevertheless, the complex spectrum of the demodulated signal can be used to determine the *locations and characteristics* of the target scatterers, and hence it can still be used for target recognition.

Example 4.4: Estimation of a radar signature.

In this example (see also [11]) we apply the adaptive FIR filtering methods to the experimental data measured by a ground-to-air radar. The target is an aircraft (specific information about the radar and the target is classified). The measured data consists of $N = 128$ samples and its quality was artificially degraded by adding zero-mean white Gaussian noise with variance 10. Figures 4.5(a)–4.5(d) show the spectral estimates of the demodulated signal obtained with DFT, windowed DFT (using a Kaiser window and shape parameter 4), Capon (with $M = 64$), and APES (with $M = 64$). The latter three methods give much lower sidelobes than the DFT method. Note also that the windowed DFT method gives wider peaks than the Capon and APES methods. ■

(a) FFT or APES with $M = 1$.

(b) FFT with Kaiser window and shape parameter 4.

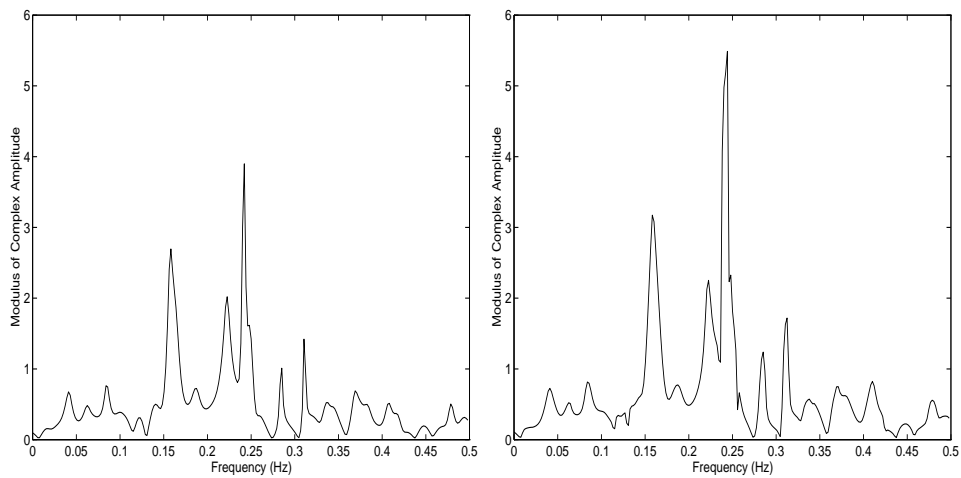
(c) Capon with $M = 64$.(d) APES with $M = 64$.

Figure 4.5. Results for Example 4.4. Spectral (range signature) estimates of an airplane observed by a ground-to-air radar using $N = 128$ samples.

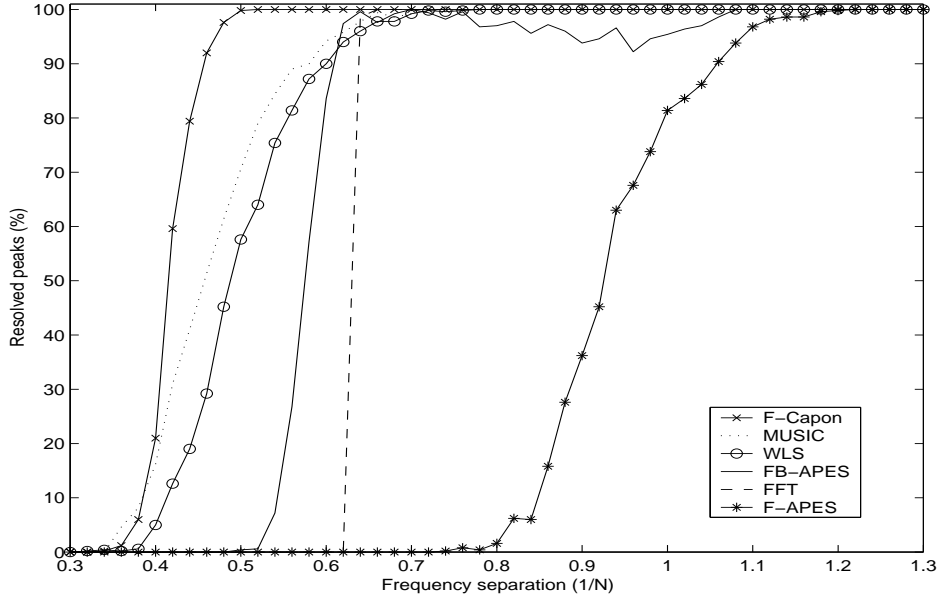


Figure 4.6. Results for Example 4.5. The figure shows the probability with which the different estimators resolve two closely spaced spectral peaks.

If $\hat{\alpha}(\omega)$ is the spectral estimate (obtained by any of the estimators above), we say that two spectral lines at frequencies ω_1 and ω_2 are resolved if

$$\left| \hat{\alpha}\left(\frac{\omega_1 + \omega_2}{2}\right) \right| < \frac{1}{2}(|\hat{\alpha}(\omega_1)| + |\hat{\alpha}(\omega_2)|). \quad (4.9.5)$$

The next example (see also [27]) demonstrates the difference in spectral resolution between Capon, APES and its FB-averaged variants.

Example 4.5: Resolution of Capon and APES.

We study the capabilities of APES, Capon, DFT and the MUSIC algorithms [8], [5] for resolving two sinusoids located at ω_1 and ω_2 , respectively. The signal under consideration is $N = 64$ samples long and consists of three dominant spectral lines (with unit amplitude) at $\omega_1 = 0.0625$, $\omega_2 = 0.0725$ and $\omega_3 = 0.25$ and ten smaller spectral lines at frequencies 0.28, 0.33, 0.35, 0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49 embedded in noise with variance 0.01.

Figure 4.6 shows the probability that the two spectral lines are resolved, as a function of the frequency separation $|\omega_1 - \omega_2|$. Clearly, F-Capon has the best resolution capability, followed closely by the MUSIC and the “WLS” estimator⁵.

⁵This estimator is *not* equivalent to the WLS estimators in Section 4.3, and it is not described herein. See [27] for details on it.

FB-APES is found to have slightly higher resolution than the DFT, whereas F-APES has a lower resolution than the DFT. In the example, the filter length in the nonparametric methods was chosen to be $M = 24$. ■

Example 4.6: Performance analysis, forward-only, 1D case.

The next example (see also [11]) considers the effect of M , the length of the FIR filter, on the accuracy of the complex amplitude estimates at frequencies that correspond to the four dominant lines in Example 4.1. The empirical root-mean-squared errors (RMSEs) of the complex amplitude estimates were obtained from 100 independent Monte-Carlo trials. Figure 4.7 shows the RMSEs of the complex amplitude estimates as a function of M . This figure also shows the CRB associated with this estimation problem (see [11] for details) which is a measure of the best performance that can be achieved by an unbiased estimator.

We can make a number of observations from Figure 4.7. Apparently, when $M = 1$ the FIR filtering approaches reduce to the FFT method and hence the performances of the various methods coincide. For some carefully picked values of M , such as $M = 16$ and $M = 28$, the RMSEs obtained with Welch and Capon are similar to those obtained with APES. Yet for such values of M , the Welch and Capon methods yield wide spectral peaks and high sidelobes and therefore their use with such relatively low values of M may not be advisable. The RMSEs of the estimates obtained with the APES algorithm are close to the CRBs for a wide range of values of M and in contrast to Welch and Capon, the selection of M in APES for accurate amplitude estimation appears to be quite easy. Since $M = N/2 = 64$ gives the narrowest spectral peaks, this example indicates that the best choice of M for the APES algorithm is $M = N/2$. ■

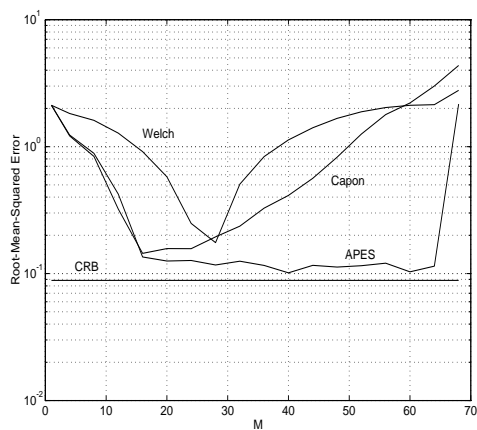
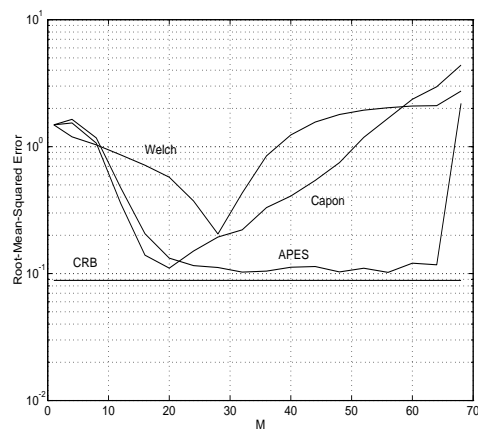
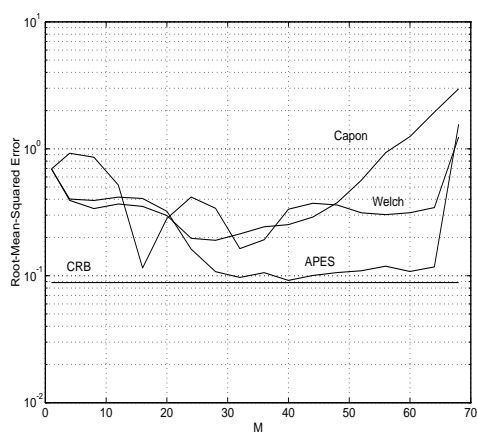
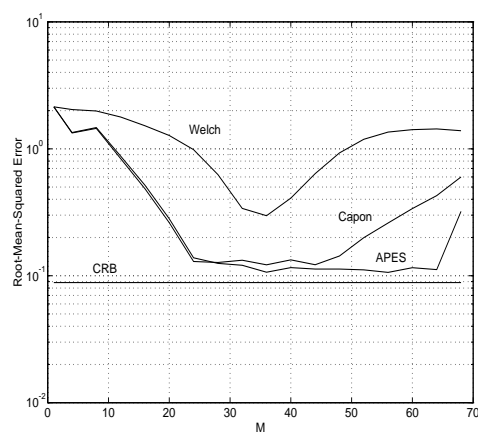
(a) Spectral line at $f = 0.04$.(b) Spectral line at $f = 0.05$.(c) Spectral line at $f = 0.08$.(d) Spectral line at $f = 0.09$.

Figure 4.7. Results for Example 6. The figure shows the RMSEs of the complex amplitude spectrum estimates obtained with Welch, Capon, and APES as a function of M for the problem considered in Example 1.

Example 4.7: Performance analysis, forward-backward averaging.

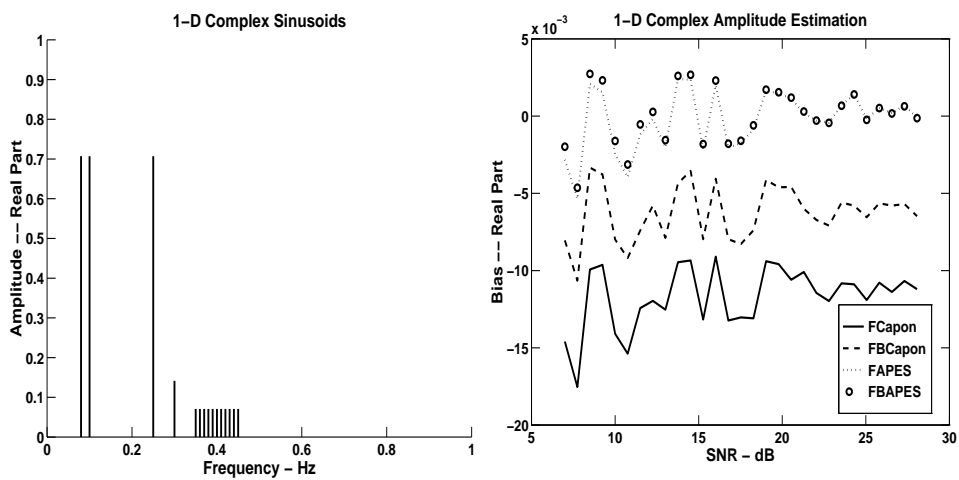
In this example (see also [21]) we compare the bias and variance of Capon, APES and their FB-averaged versions for an 1D data set of length $N = 64$ consisting of 15 superimposed complex sinusoids embedded in zero-mean Gaussian noise. The real part of the amplitude spectrum is shown in Figure 4.8(a). The amplitude spectrum's imaginary part is identical to its real part and therefore is not shown here.

We begin by studying the performance of the estimators as the signal-to-noise ratio (SNR) varies. The SNR for the k th sinusoid is defined as

$$\text{SNR}_k = 10 \log_{10} \left(\frac{|\alpha_k|^2}{P(\omega_k)} \right) \quad (\text{dB}) \quad (4.9.6)$$

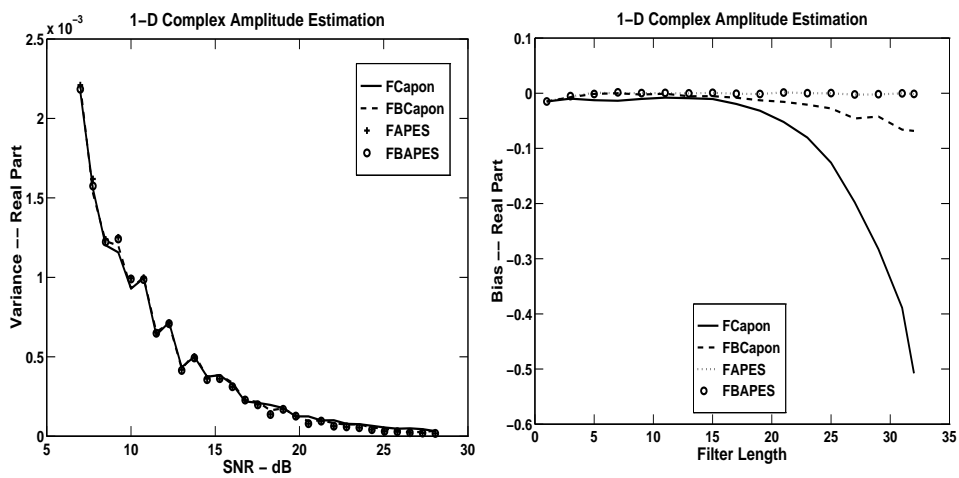
where α_k is the complex amplitude of the k th sinusoid and $P(\omega_k)$ is the spectral density of the additive noise at frequency ω_k . The filter length is chosen as $M = 15$. The real parts of the biases, as a function of SNR_1 , are shown in Figure 4.8(b). The imaginary parts behave similarly and are therefore not shown. As seen from these figures, F-APES and FB-APES are almost unbiased, while F-Capon and FB-Capon are biased downward. In addition, we notice that the bias of F-Capon is approximately twice that of FB-Capon. All these observations are consistent with Theorem 4.2. The variances of the real parts of the amplitude estimates are shown in Figure 4.8(c) (the imaginary parts behave similarly and are not shown). For this example, all of the estimators have similar variances. However, as we will see shortly, the variance of Capon becomes notably larger than that of APES as M increases.

We next study the performance as the filter length M varies for a fixed SNR_1 equal to 20 dB. Figure 4.8(d) shows the real parts of the biases for some different M , and Figure 4.8(e) shows the corresponding variance. From Figures 4.8(d) and 4.8(e), we can see that both F-APES and FB-APES are unbiased for all practical filter lengths, whereas the bias of Capon grows significantly with increasing M . In fact, all filterbank methods reduce to the Fourier transform approach when $M = 1$, and only when M is sufficiently large the filterbank approach shows a noticeable improvement over the Fourier method. All estimators seem to perform similarly for M up to a fourth of the data length, with Capon being slightly biased downward. As the filter length increases further, the performance of Capon degrades rapidly, while that of APES remains unaffected. ■



(a) True spectrum.

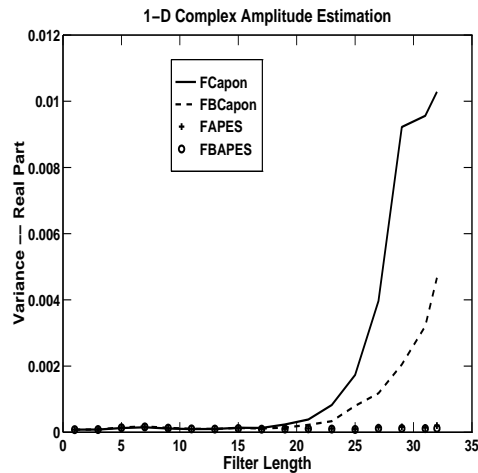
(b) Bias of real part.



(c) Variance of real part.

(d) Bias of real part.

Figure 4.8. (continued on the next page)



(e) Variance of real part.

Figure 4.8. (cont.) Results for Example 4.7.

To summarize, we have found that as M increases, all the estimators under study achieve better spectral resolution and that the best resolution is obtained at $M = N/2$. This fact, along with the statistical results shown in the previous examples, indicates that the choice of M for Capon should be done by a tradeoff between resolution and statistical stability. Usually we choose $N/4 \leq M \leq N/2$. While the choice of M for Capon is difficult to make, it is easy to see that APES achieves the best performance at $M = N/2$, since with this choice, APES achieves the highest possible resolution as well as the best statistical properties in terms of bias and variance. The previous examples also show that F-APES and FB-APES perform similarly in terms of bias and variance properties for the frequency of interest, although FB-APES has a slightly better resolution than F-APES.

4.9.2 2D Examples

We next show some examples of 2-D spectral estimation.

Example 4.8: Estimation of a 2D complex line spectrum.

In this example we consider a 2D data matrix of size 32×32 consisting of three 2D sinusoids with frequencies $(1, 0.8)$, $(1, 1.1)$, $(1.1, 1.3)$ and amplitudes 1, 0.7 and 2, respectively, embedded in white Gaussian noise with standard deviation 0.1. In Figure 4.9(a) we show the true spectrum. Figure 4.9(b) shows a windowed Fourier

transform spectrum (WFT) of the data evaluated on a 512×512 grid. The WFT is the Fourier transform of the data multiplied with a radially symmetric 2D-Taylor window with shape parameter 5 and sidelobe level -35 dB. Figures 4.9(c) and 4.9(d) show the APES and Capon spectra evaluated using a filter matrix of size 8×8 . In Figure 4.9(e) and 4.9(f), the corresponding results for a filter matrix of size 14×14 are shown. Note the superior resolution of Capon and APES compared with the windowed DFT. Also note that with a filter size of 14×14 , the Capon spectral estimate becomes somewhat noisy. ■

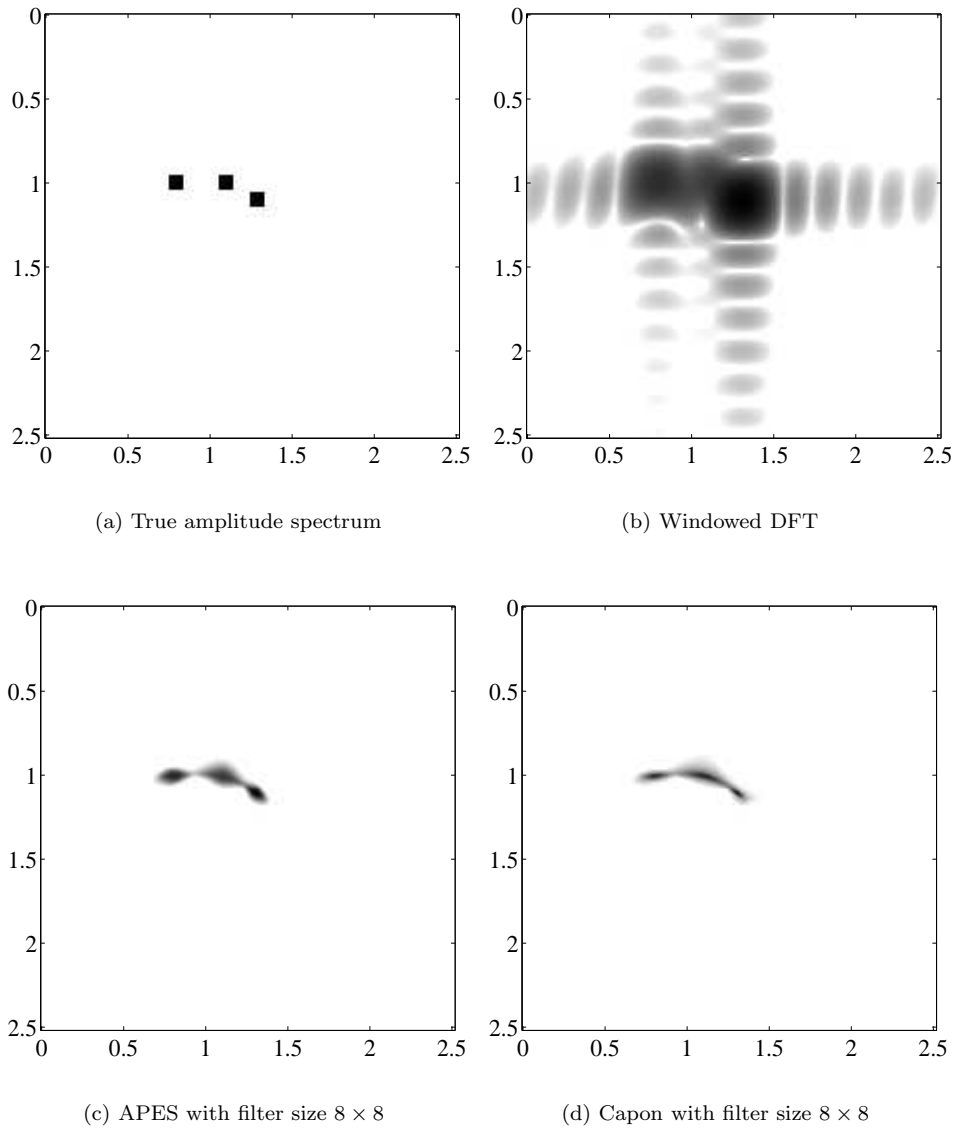


Figure 4.9. (continued on the next page)

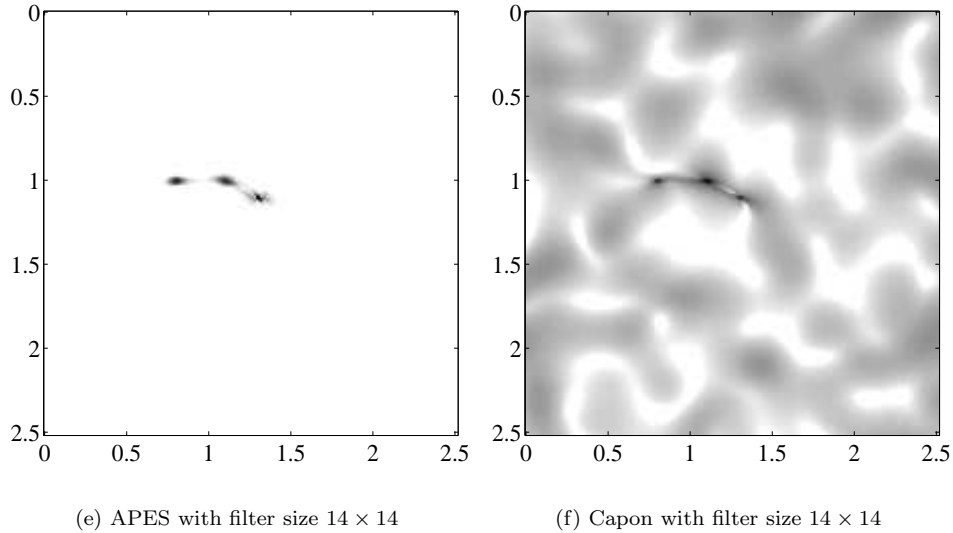


Figure 4.9. (cont.) Results for Example 4.8. Spectral analysis of a 2D data matrix with adaptive filterbank methods. The dynamic range is 40 dB (amplitude) in all figures.

It can be shown that a 2D DFT is equivalent to first performing a row-wise 1D DFT and subsequently a column-wise 1D DFT, i.e., applying a 1D DFT to each row of the data matrix first and then a 1D DFT to each column of the resulting intermediate matrix. Since the adaptive filterbank spectra are nonlinear functions of the data, this result does not in general carry over to the case of Capon and APES. Therefore, it can be interesting to compare the quality of the “true” 2D spectral estimates with that of the estimates obtained via repeated 1D processing. Specifically, we can compare the result of applying a row-wise 1D Capon or APES estimator to the data and then use column-wise 1D Capon/APES on the so-obtained intermediate matrix with the result of doing the operations in reverse order (i.e., column-wise processing before row-wise processing). Depending on whether the rows or the columns are processed first, two different estimates $\alpha(\omega, \bar{\omega})$ are obtained. It was found in [11] that taking the spectral estimate equal to the minimum absolute value (in a fashion reminiscent of spatially variant apodization [38]) appears to give the best result that a 1D approach can achieve. The so-obtained approach was called M-APES and M-Capon, respectively. In the next example, we will compare the 1D, 2D and M variants of Apes and Capon for a 2D signal that has a spectrum consisting of continuous components and of spectral lines.

Example 4.9: 1D vs. 2D vs. M Capon and APES for 2D data.

In this example (see also [11]) we consider a data matrix of size $N \times \bar{N} = 32 \times 32$ consisting of 3 spectral lines (whose magnitudes are 5, 7, and 10) and two closely-spaced one-dimensional continuous pulses (whose magnitudes are 6.15). The data are corrupted by additive zero-mean white Gaussian noise with variance 20. The true spectrum is shown in Figure 4.10(a). The modulus (in dB) of the estimated spectrum is shown in Figures 4.10(b)–4.10(e), using different estimation techniques. Figure 4.10(b) shows the modulus of the estimate obtained with the 2D DFT method. The use of the 2D DFT results in high sidelobes; the two 1D continuous pulses in the spectrum are barely resolved. Figure 4.10(c) shows the modulus of the spectral estimate obtained by using a 2D DFT with a circularly symmetric Kaiser window with shape parameter 4. Whilst the windowed DFT method reduces the sidelobes of the DFT, it widens the spectral peaks and consequently the closely spaced one-dimensional continuous pulses are smeared together. Figures 4.10(d) and 4.10(e) show the SAR image estimates obtained by using the 2D-Capon (with $M = \bar{M} = 16$) and 2D-APES (with $M = \bar{M} = 16$), respectively. Note again that 2D-APES gives slightly wider spectral peaks than 2D-Capon, but the spectral estimates obtained with the former are more accurate.

In Figure 4.10(f) we show the modulus of the spectral estimate obtained by applying the 1D APES method with $M = 16$ first to the rows and then to the columns of the data matrix. Although the 1D APES method has a better resolution than the windowed DFT, the spectral estimate obtained with the former contains large vertical sidelobes. On the other hand, when the 1D APES method is applied first to the columns and then to the rows of the data matrix, large horizontal sidelobes occur in the estimate (this result is not shown here). In Figures 4.10(g) and 4.10(h) show the estimates obtained by using the M-Capon (with $M = 16$) and M-APES (with $M = 16$) methods, respectively. Clearly, M-APES is better than 1D APES. Note also that M-APES gives somewhat wider 2D spectral peaks than M-Capon, but the former yields more accurate spectral estimates than the latter, particularly so for the continuous parts of the image. Finally, comparing Figures 4.10(g) and 4.10(h) with Figures 4.10(d) and 4.10(e), we see that the 2-D methods give better estimates of the amplitude spectrum than the M-approaches. This is an expected result: 2D processing should be superior to 1D processing. ■

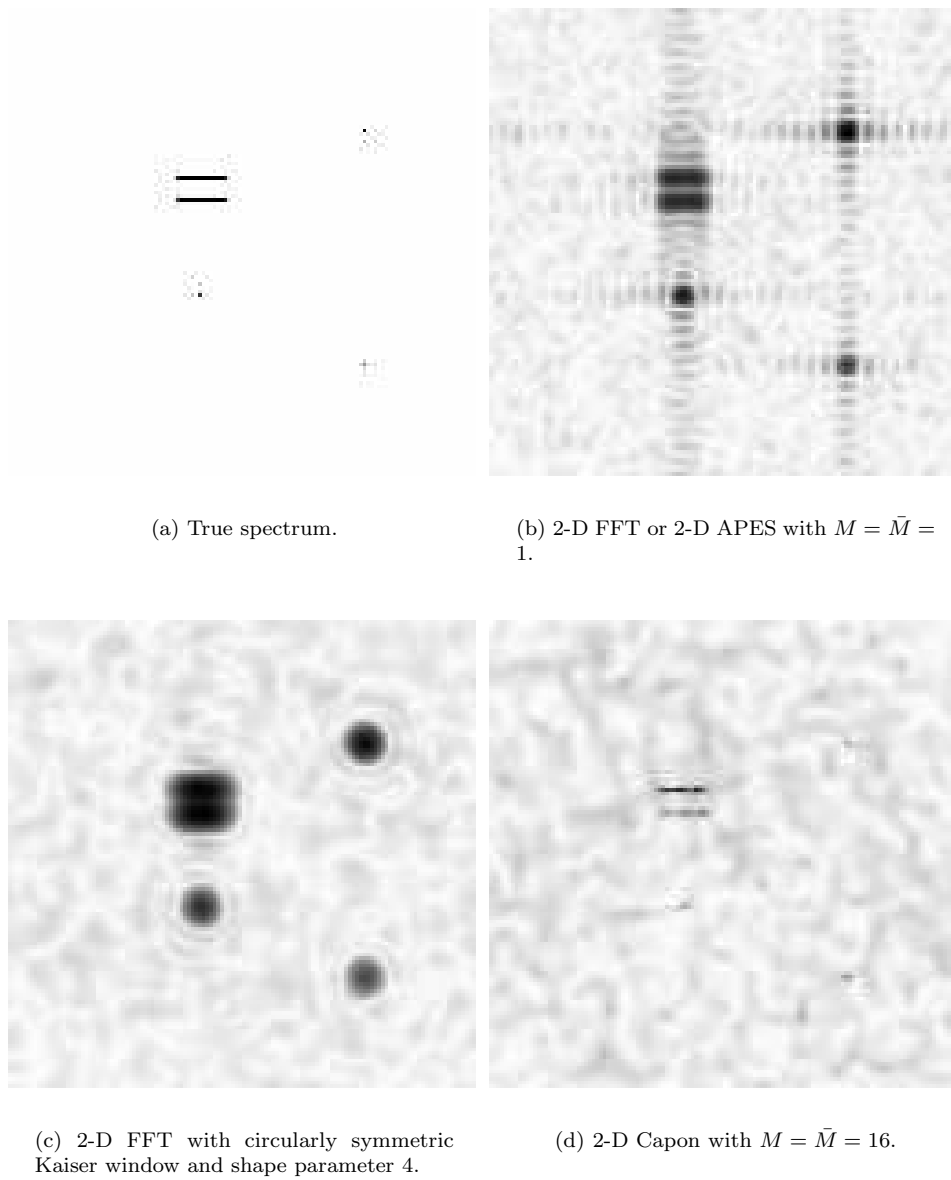
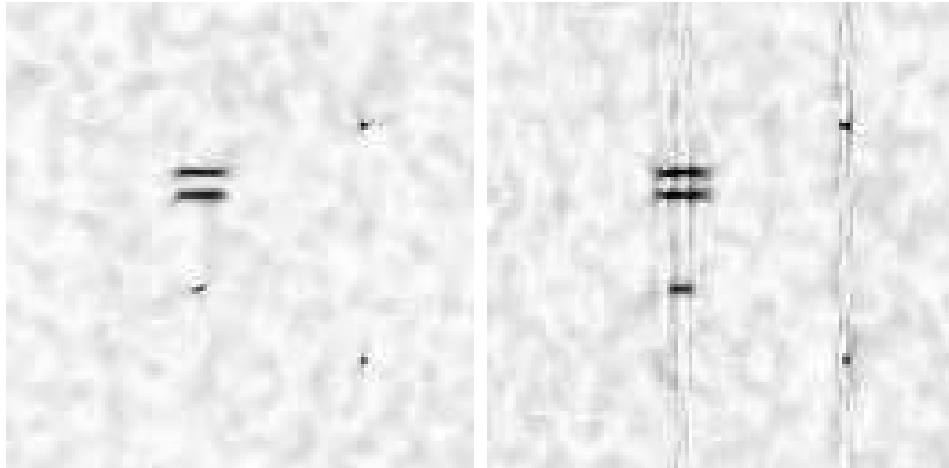
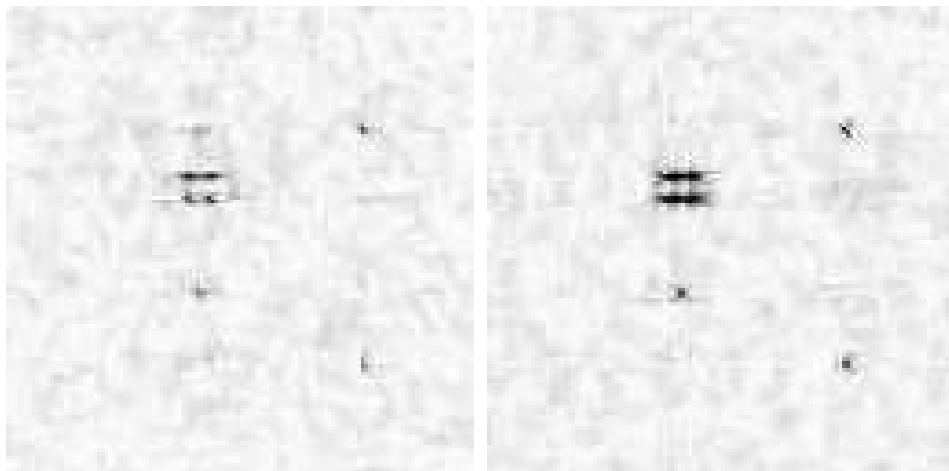


Figure 4.10. (continued on the next page)

(e) 2-D APES with $M = \bar{M} = 16$.(f) 1-D APES (rows-then-columns) with $M = 16$.(g) M-Capon with $M = 16$.(h) M-APES with $M = 16$.**Figure 4.10.** (cont.) Results for Example 4.9.

4.9.3 Synthetic Aperture Radar (SAR) Imaging Examples

We next consider using the FIR filtering approaches for SAR imaging. In SAR or inverse SAR (ISAR) imaging, the radar usually transmits linear FM chirp pulses (like in target signature estimation applications) [39], [40]. Upon receiving each pulse reflected by the object being imaged, the radar demodulates the pulse by mixing it with a reference chirp signal in the same way as described above for range signature estimation. As a result, the scattering centers of the object at different ranges correspond to different frequencies of the demodulated signal. Since either the radar or the object is moving or rotating, the pulses received at different angles between the radar and the object are used to form a synthetic aperture. After a Polar-to-Cartesian frequency interpolation, the scattering centers of the object at the same range but different cross-ranges correspond to different (Doppler) frequencies over the synthetic aperture.

The 2D signal obtained with a synthetic aperture radar is referred to as the SAR phase history, and takes on the following form:

$$X(n, \bar{n}) = \int \alpha(\omega, \bar{\omega}) e^{-i(\omega n + \bar{\omega} \bar{n})} d\omega d\bar{\omega} \quad (4.9.7)$$

where $\alpha(\omega, \bar{\omega})$ is the object reflectivity function. In the “ideal” case that the object consists of point-scatterers only, (4.9.7) reduces to a sinusoidal model similar to that in (4.2.2):

$$X(n, \bar{n}) = \sum_{p=1}^P \alpha_p e^{-i(\omega_p n + \bar{\omega}_p \bar{n})}. \quad (4.9.8)$$

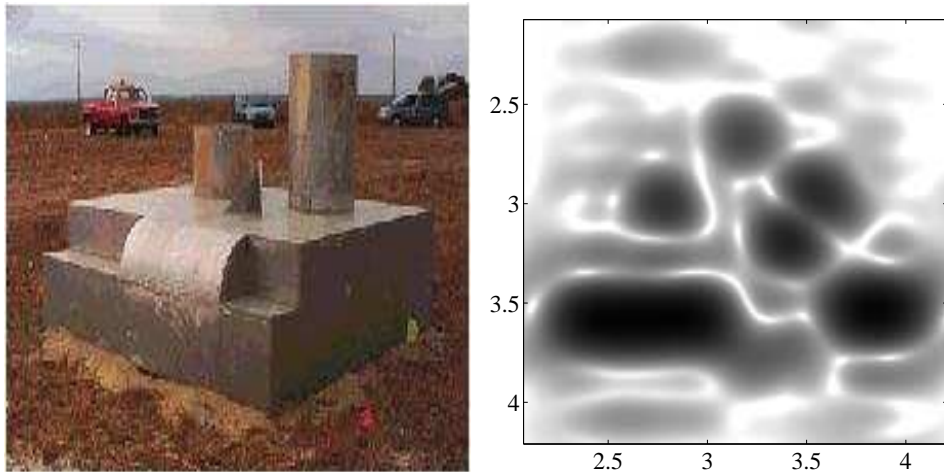
From these equations it follows that the task of forming a SAR image from the phase history data is essentially a 2D spectral estimation problem.

The conventional SAR imaging methods include the DFT or windowed DFT methods. Many parametric and nonparametric spectral estimation methods have also more recently been used for SAR imaging [3], [41], [42], [43], [44]. It has been shown in [3] that the minimum variance method (which is similar to the Capon method herein) gives good SAR images while the high resolution parametric methods advocated in some of the cited references tend to be too sensitive to model errors. In the next example, we will apply the APES and Capon algorithms to SAR imaging.

Example 4.10: SAR imaging of a slicy object.

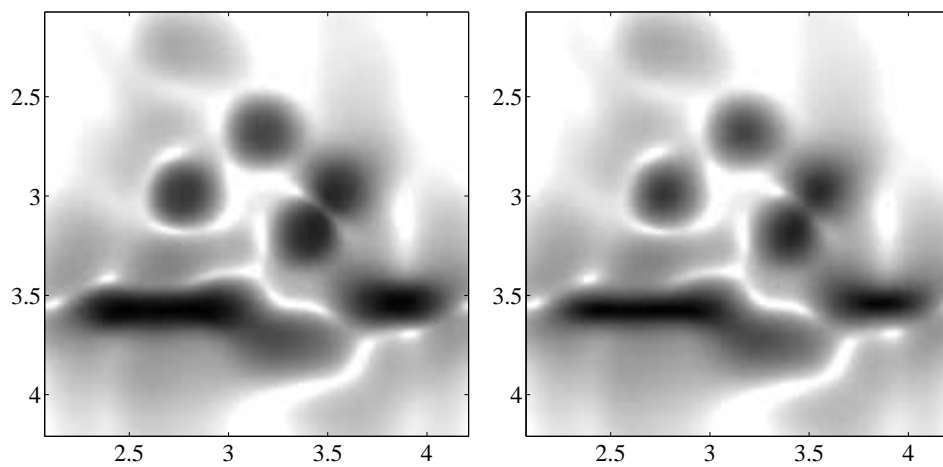
We consider a 32×32 matrix of phase history data of an object (at 0° azimuth angle) generated by XPATCH, a high frequency electromagnetic scattering prediction code for complex 3D-objects [45]. A photo of the object under consideration (taken at 45° azimuth angle) is shown in Figure 4.11(a). In Figure 4.11(b) we show the SAR image obtained by applying a windowed Fourier transform (WFT) to the data (which is a standard method for SAR imaging). The same 2D window as in Example 8 was used. Figures 4.11(c) and 4.11(d) show the SAR image obtained via application of

APES and Capon using a filter matrix of size 8×8 . In Figures 4.11(e) and 4.11(f), the corresponding results for a filter matrix of size 14×14 are shown. It is clear that Capon and APES outperform the windowed DFT, and also that the increase in filter lengths improves the resolution of the SAR image significantly. ■



(a) Photograph of the object (at 45° azimuth angle)

(b) SAR image obtained via windowed DFT



(c) SAR image obtained via APES with filter size 8×8

(d) Capon with filter size 8×8

Figure 4.11. (continued on the next page)

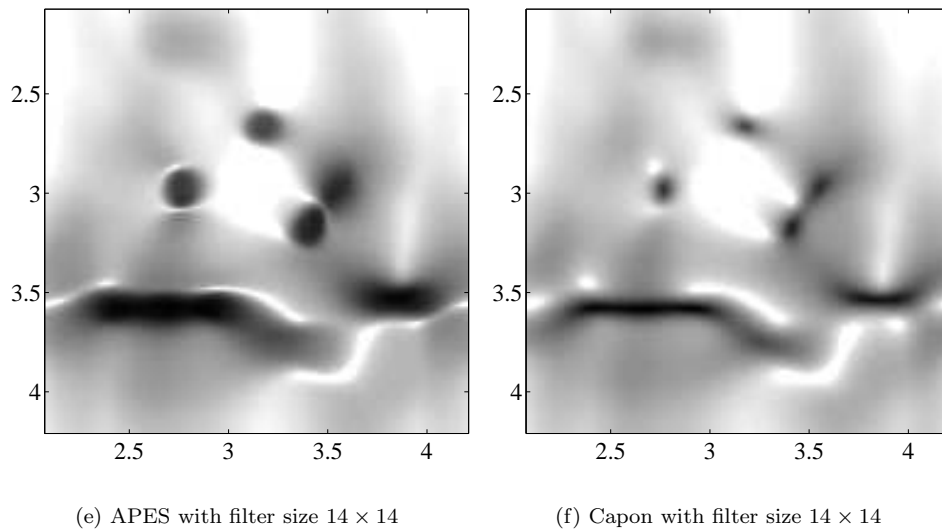
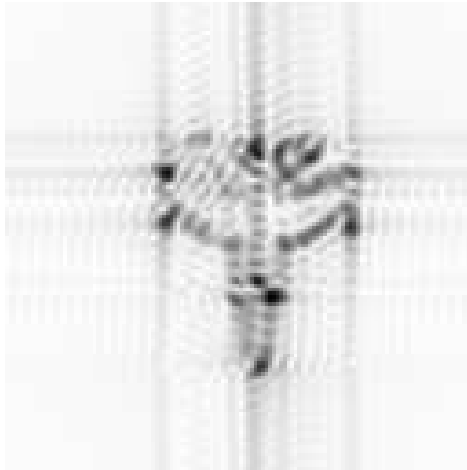


Figure 4.11. (cont.) Results from Example 4.10. SAR imaging of a “slicy” object via adaptive filterbank methods. The dynamic range is 35 dB (amplitude) in all sub-figures.

Example 4.11: ISAR imaging of a MIG-25 aircraft.

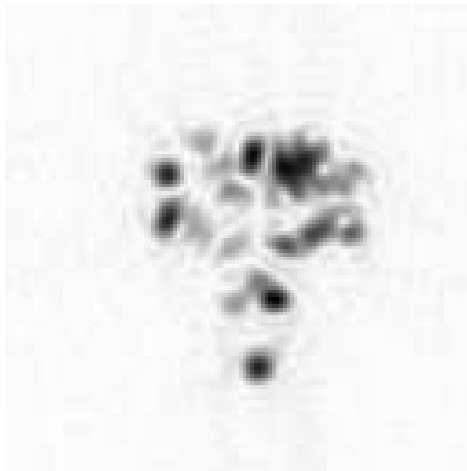
We consider inverse SAR imaging of two snapshots of a simulated fast rotating MIG-25 aircraft. The data matrices are of size 32×32 and were provided by the Naval Research Laboratory. Figures 4.12(a) and 4.12(b) show the result obtained by 2D DFT, and Figures 4.12(c) and 4.12(d) show the result obtained by a windowed 2D DFT. Finally, Figures 4.12(e) and 4.12(f) show the result of applying 2D APES. Note again that APES outperforms the DFT and windowed DFT methods. ■



(a) 2D DFT image, snapshot I



(b) 2D DFT image, snapshot II



(c) Windowed 2D DFT image, snapshot I



(d) Windowed 2D DFT image, snapshot II

Figure 4.12. (continued on the next page)

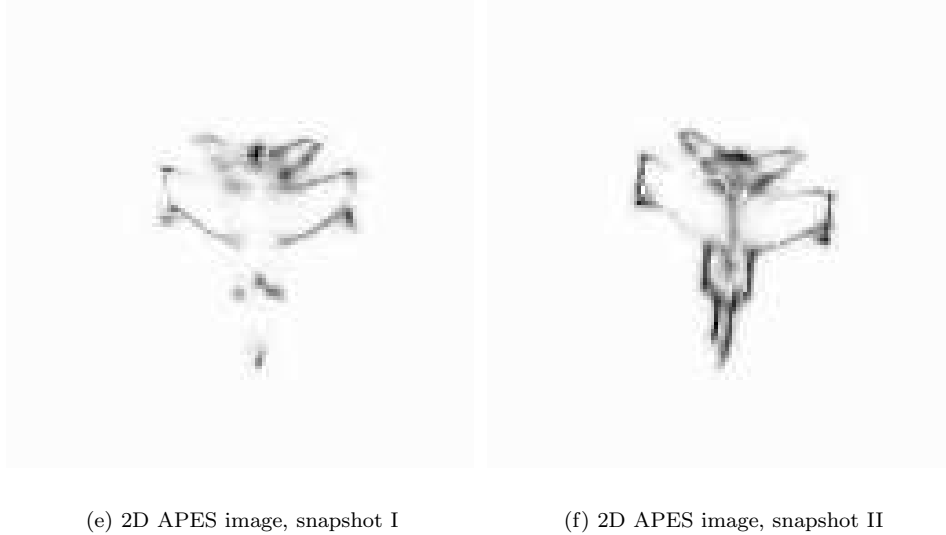


Figure 4.12. (cont.) Results from Example 4.11.

4.10 Nonparametric Spectral Analysis of Gapped Data

In recent years, the problem of performing spectral analysis of incompletely sampled data has attracted some attention. In this case, the signal under consideration contains missing samples or segments. Apart from being of theoretical interest, this problem is of considerable importance in a number of applications ranging from SAR imaging with angular diversity to astronomical time-series analysis. The missing samples are often, but not necessarily, clustered together forming gaps in the collected data. In some applications it is common that entire rows or columns of the data matrix are missing.

In time-series analysis applications it is common that observations are unavailable during certain periods of time, and therefore the observed data contain gaps [46]. This can be so due to sensor failure, or due to an inherent inability to observe the physical quantity of interest. In a SAR system that employs angular diversity, a target is illuminated by a radar from several separate angular (cross-range) regions independently. This means that the radar measurements are taken discontinuously as the platform travels along its trajectory (see Figure 4.13), and consequently the resulting phase-history matrix will have missing columns. Since the radar is available for other tasks during the “idle” periods, the angular diversity technique can reduce the amount of radar resources needed for the imaging of a particular target while maintaining the same cross-range aperture as if measurements were taken



Figure 4.13. SAR imaging with angular diversity.

continuously. In both these applications, the locations of the missing samples are always known.

A simple way to estimate the spectrum in the case of missing data would be to compute the conventional DFT spectrum with the missing data set to zero. However, it is well-known that this usually gives rise to an estimated spectrum with serious artifacts. It is not hard to understand why: setting the missing samples to zero corresponds to multiplying the original time-series with a windowing function that assumes a value of one whenever a sample is available, and zero otherwise. In the frequency-domain, this procedure carries over to a convolution between the Fourier transform of the signal and that of the windowing function. Since the Fourier transform of the windowing function typically has very large sidelobes in the gapped-data case, the resulting spectral estimate will contain significant artifacts.

To deal with the sidelobe problem, and to overcome the resolution limit of the DFT, several sophisticated methods for spectral analysis of gapped data have appeared in the literature. The so-called CLEAN algorithm [47] estimates the spectrum by fitting and removing sinusoids to and from the data in an iterative fashion. It can be shown that under certain conditions, CLEAN essentially computes the solution to a deconvolution problem [48] and consequently the CLEAN spectrum can be evaluated efficiently using suitable numerical methods [49]; however, it is known that CLEAN usually experiences problems when trying to resolve closely spaced spectral lines. The so-called multi-taper methods (see, e.g., [50], [51]) compute spectral estimates that are quadratic forms of the data. The coefficients in the corresponding quadratic functions are optimized according to certain criteria, and depend only on the gapping pattern. It appears that these algorithms cannot overcome the resolution limit of the DFT. Finally, there are methods that attempt

to fit an autoregressive (AR) model to the data and use this model to interpolate the gaps. Doing so may well fail since the AR generally is strictly stable, and hence the extrapolated samples decay to zero as the length of a gap grows.

In this section, we describe two nonparametric methods for spectral estimation of gapped data. We start in Subsection 4.10.1 by presenting a modification of the WLS technique in Section 4.3 to gapped data, and to *periodically* gapped data in particular. The method described in Subsection 4.10.1 is interpolation-free, which may have certain advantages (in particular for computational reasons), and appeared first in [52]. Next, in Subsection 4.10.2 we go on to describe an extension of the APES method to gapped data, called gapped-data APES (GAPES). The GAPES method is an interpolation-based algorithm that can deal with quite general sampling patterns, and was developed in a series of papers [53], [54], [36].

4.10.1 Spectral Estimation of Gapped Data via WLS

We can obtain a WLS formulation (similar to that in Section 4.3) for the problem of spectral analysis of incomplete data. Following [52] suppose that $x(n)$ is available only for certain time instants. Let

$$\mathcal{A} = [n_0, \dots, n_{N_a-1}] \subset [0, \dots, N-1] \quad (4.10.1)$$

denote the index set of the N_a sample indices for which $x(n)$ is available, and let $\mathbf{y}_{\mathcal{A}}$ be a vector that contains the samples in \mathcal{A} . If we can form snapshots $\{\mathbf{y}_l\}$ of a given length M in a similar way as for uniformly sampled data, and such that the corresponding residual vectors $\{\mathbf{w}_l(\omega)\}$ are identically distributed (but not necessarily independent), then the WLS criterion (4.3.9) is in principle applicable directly.

The article [52] suggested to construct snapshots as follows:

$$\mathbf{y}_l = [x(k_l + m_0) \ x(k_l + m_1) \ \dots \ x(k_l + m_{M-1})]^T \quad (4.10.2)$$

for $l = 0, \dots, L-1$, where $\{k_l\}_{l=0}^{L-1}$ and $\{m_r\}_{r=0}^{M-1}$ are given sets of numbers. Then provided that $w(n)$ is stationary, the residual vectors associated with \mathbf{y}_l will be identically distributed (although not independent). Let

$$\begin{aligned} \check{\mathbf{A}}(\omega) &= \check{\mathbf{a}}_M(\omega) \check{\mathbf{a}}_L^T(\omega) \\ \check{\mathbf{a}}_M(\omega) &= [e^{i\omega m_0} \ \dots \ e^{i\omega m_{M-1}}]^T, \\ \check{\mathbf{a}}_L(\omega) &= [e^{i\omega k_0} \ \dots \ e^{i\omega k_{L-1}}]^T \end{aligned} \quad (4.10.3)$$

Since

$$\mathbf{y}_l = \alpha(\omega) \check{\mathbf{a}}_M(\omega) e^{i\omega k_l} + \mathbf{w}_l(\omega) \quad (4.10.4)$$

and because $\{\mathbf{w}_l(\omega)\}$ are identically distributed, we can formulate a WLS criterium similar to that in Section 4.3:

$$\hat{\alpha}(\omega) = \underset{\alpha(\omega)}{\operatorname{argmin}} \left\| \Phi^{-1/2}(\omega) (\mathbf{Y} - \alpha(\omega) \check{\mathbf{A}}(\omega)) \right\|^2. \quad (4.10.5)$$

The solution to (4.10.5) is

$$\hat{\alpha}(\omega) = \frac{\check{\mathbf{a}}_M^H(\omega)\mathbf{\Phi}^{-1}(\omega)\check{\mathbf{g}}(\omega)}{\check{\mathbf{a}}_M^H(\omega)\mathbf{\Phi}^{-1}(\omega)\check{\mathbf{a}}_M(\omega)} \quad (4.10.6)$$

where

$$\check{\mathbf{g}}(\omega) \triangleq \frac{1}{L}\mathbf{Y}\check{\mathbf{a}}_L^*(\omega). \quad (4.10.7)$$

By choosing different $\mathbf{\Phi}(\omega)$, both APES and Capon-like methods can be designed and applied in a relatively straightforward way. A Capon-like method is obtained by choosing

$$\mathbf{\Phi}(\omega) = \frac{1}{L}\mathbf{Y}\mathbf{Y}^H \quad (4.10.8)$$

and an APES-like method is obtained by taking

$$\mathbf{\Phi}(\omega) = \frac{1}{L}\mathbf{Y}\mathbf{Y}^H - \check{\mathbf{g}}(\omega)\check{\mathbf{g}}^H(\omega). \quad (4.10.9)$$

In the degenerate case of $M = 1$, we can simply take $\mathbf{y}_l = x(n_l)$ for $l = 0, \dots, N_a - 1$ (in this case, $\{\mathbf{y}_l\}$ become scalars). It is easy to verify that the application of the WLS (which corresponds to the DFT in the case of uniformly sampled data), now corresponds to applying a DFT to $x(n)$ but with the missing samples set to zero:

$$\hat{\alpha}_{\text{G-DFT}}(\omega) = \frac{1}{N_a} \sum_{n \in \mathcal{A}} x(n)e^{-i\omega n}. \quad (4.10.10)$$

As already explained, this gapped-DFT (G-DFT) spectrum usually contains significant artifacts. In particular, if $x(n)$ is a single sinusoid with unit amplitude embedded in noise as in Subsection 4.7.2 above, the G-DFT spectrum (for high SNR) is essentially equal to the DFT of the windowing function induced by the gapping pattern:

$$\hat{\alpha}_{\text{G-DFT}}(\omega) \rightarrow \frac{1}{N_a} \sum_{n \in \mathcal{A}} e^{i(\omega_0 - \omega)n}, \quad \sigma^2 \rightarrow 0 \quad (4.10.11)$$

which can take on a more or less arbitrary shape. For certain gapping patterns, for instance periodic, this problem is so significant that the G-DFT spectrum estimate is useless even for $|\omega - \omega_0| \gg 2\pi/N_a$. Again, the artifacts in the spectrum can be attributed to the non-orthogonality between $\check{\mathbf{a}}_N(\omega_0)$ and $\{\check{\mathbf{a}}_N(\omega)\}$ for $\omega \neq \omega_0$.

In the general case of $M > 1$, the leakage effects associated with the estimate in (4.10.6) can be assessed by an analysis similar to that in Subsection 4.7.2. Let us consider the Capon-like estimator, i.e., choose $\mathbf{\Phi}(\omega) = (1/L)\mathbf{Y}\mathbf{Y}^H$. If $x(n)$ consists of a sinusoid with frequency ω_0 in noise as in Subsection 4.7.2, then for high SNR

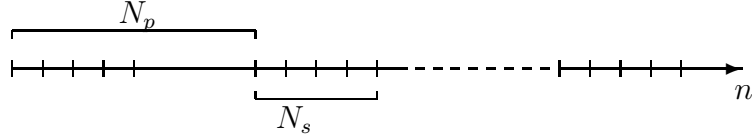


Figure 4.14. Periodic gapped-data sampling. There are N_c clusters of samples, each of length N_p . Each cluster contains N_s available samples and $N_p - N_s$ missing samples.

and a reasonably large number of snapshots, a calculation similar to that in (4.7.7) shows that

$$\sigma^2 \Phi^{-1}(\omega) = \sigma^2 L \cdot (\mathbf{Y}\mathbf{Y}^H)^{-1} \approx \Pi_{\check{\mathbf{a}}_M(\omega_0)}^\perp. \quad (4.10.12)$$

Provided that $M > 1$ we have that:

$$\hat{\alpha}(\omega) \approx \frac{\check{\mathbf{a}}_M^H(\omega) \Pi_{\check{\mathbf{a}}_M(\omega_0)}^\perp \check{\mathbf{g}}(\omega)}{\check{\mathbf{a}}_M^H(\omega) \Pi_{\check{\mathbf{a}}_M(\omega_0)}^\perp \check{\mathbf{a}}_M(\omega)} \approx \frac{1}{L} \frac{\check{\mathbf{a}}_M^H(\omega) \Pi_{\check{\mathbf{a}}_M(\omega_0)}^\perp \check{\mathbf{a}}_M(\omega_0) \check{\mathbf{a}}_L^T(\omega_0) \check{\mathbf{a}}_L^*(\omega)}{\check{\mathbf{a}}_M^H(\omega) \Pi_{\check{\mathbf{a}}_M(\omega_0)}^\perp \check{\mathbf{a}}_M(\omega)} = 0 \quad (4.10.13)$$

for $\omega \neq \omega_0$ and high SNR. Hence we can expect the methods based on (4.10.6) to be essentially leakage-free, even though the data sequence contains gaps.

Periodically Gapped Data

The key problem is of course how the snapshots, or rather, the parameters $\{k_l\}$ and $\{m_l\}$ in (4.10.2), should be chosen. In general, the selection of snapshots appears to be a hard problem that is open. However, for some important special cases, including that of *periodically* gapped data, the snapshots can be chosen in a systematic manner.

The case of periodically gapped data is illustrated in Figure 4.14. In this case, we assume that the gapping pattern is periodic and that data are available at the time instants:

$$\begin{aligned} n = 0, \dots, N_s - 1, \\ N_p, \dots, N_p + N_s - 1, \\ 2N_p, \dots, 2N_p + N_s - 1, \\ \dots \\ (N_c - 1)N_p, \dots, (N_c - 1)N_p + N_s - 1 \end{aligned} \quad (4.10.14)$$

Here N_p is the periodicity of the gapping, N_s is the number of available samples per cluster and N_c is the number of clusters. Clearly, $N_a = N_c N_s$ samples are available in total.

Choice of Snapshots: In the case of periodically gapped data, we can construct snapshots as follows. Let \check{N}_s and \check{N}_c be two user parameters, and choose

$$\begin{aligned}
 k_l = & \{0, 1, \dots, N_s - \check{N}_s, \\
 & N_p, N_p + 1, \dots, N_p + N_s - \check{N}_s, \\
 & \dots, \\
 & (N_c - \check{N}_c)N_p, (N_c - \check{N}_c)N_p + 1, \dots, (N_c - \check{N}_c)N_p + N_s - \check{N}_s\} \\
 m_l = & \{0, 1, \dots, \check{N}_s - 1, \\
 & N_p, N_p + 1, \dots, N_p + \check{N}_s - 1, \\
 & \dots, \\
 & (\check{N}_c - 1)N_p, (\check{N}_c - 1)N_p + 1, \dots, (\check{N}_c - 1)N_p + \check{N}_s - 1\}
 \end{aligned} \tag{4.10.15}$$

in (4.10.2). Clearly, the total number of snapshots is $L = (N_s - \check{N}_s + 1) \cdot (N_c - \check{N}_c + 1)$ and their length is $M = \check{N}_c \check{N}_s$. Hence the corresponding matrix \mathbf{Y} will be of dimension $\check{N}_c \check{N}_s \times (N_s - \check{N}_s + 1)(N_c - \check{N}_c + 1)$. An equivalent way of describing the above choice of snapshots is to let $0 \leq \check{n}_s \leq N_s - \check{N}_s$, $0 \leq \check{n}_c \leq N_c - \check{N}_c$, and take snapshot number $l = \check{n}_s + \check{n}_c N_p$ (i.e., the l th column of \mathbf{Y}) as:

$$\begin{aligned}
 \mathbf{y}_l = & \begin{bmatrix} x(l) & x(l+1) & \dots & x(l + \check{N}_s - 1) \\
 & x(l + N_p) & x(l + N_p + 1) & \dots & x(l + N_p + \check{N}_s - 1) \\
 & \dots & \dots & \dots & \dots \\
 & & & & x(l + (\check{N}_c - 1)N_p) & x(l + (\check{N}_c - 1)N_p + 1) & \dots & x(l + (\check{N}_c - 1)N_p + \check{N}_s - 1) \end{bmatrix}^T
 \end{aligned} \tag{4.10.16}$$

The formation of snapshots is illustrated in Figure 4.15. By forming snapshots this way and applying (4.10.6) with $\Phi(\omega)$ chosen according to (4.10.8) or (4.10.9), we arrive at two new methods that we will call periodically-gapped Capon, respectively.

Interpretation and Fast Implementation: The procedure described above for the formation of snapshots and the related application of Capon or APES can be interpreted as forming a matrix of size $N_s \times N_c$ by stacking the available data segments on top of each other, and applying 2D Capon/APES with a matrix filter of dimension $\check{N}_s \times \check{N}_c$ to the so-obtained matrix. This interpretation is useful for providing more insight into how PG-Capon and PG-APES work, and for obtaining a fast implementation.

To understand how PG-Capon/PG-APES and 2-D Capon/2-D APES/ are related, assume for the moment, although this is not a strictly necessary assumption for Capon or APES, that $x(n)$ is a superposition of sinusoids with amplitudes $\{\alpha_k\}$ and frequencies $\{\omega_k\}$ in stationary noise $w(n)$:

$$x(n) = \sum_k \alpha_k e^{i\omega_k n} + w(n). \tag{4.10.17}$$

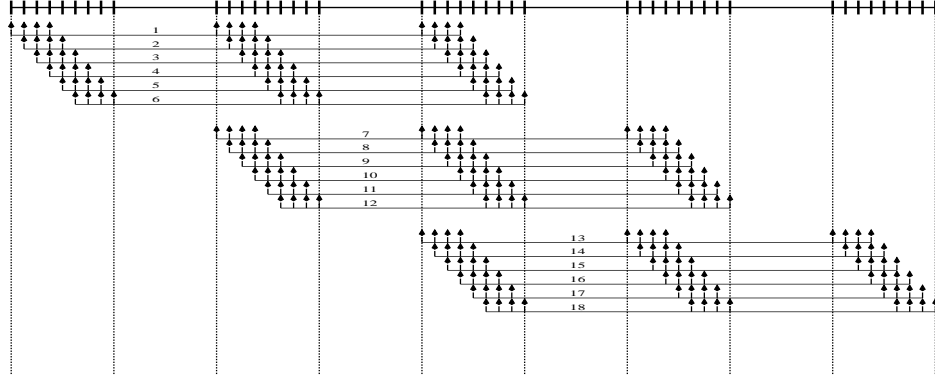


Figure 4.15. Illustration of the formation of snapshots in the case of periodically gapped data. In this example, $N_c = 5$, $N_s = 9$, $\check{N}_c = 3$, $\check{N}_s = 4$.

By stacking the available segments of $x(n)$ on top of each other, we effectively construct the following *fictitious* 2D data set:

$$X(n_s, n_c) = x(n_s + n_c N_p) \quad (4.10.18)$$

for $0 \leq n_s \leq N_s - 1, 0 \leq n_c \leq N_c - 1$. Observe from (4.10.17) that

$$X(n_s, n_c) = \sum_k \alpha_k e^{i(\omega_k n_s + \omega_k n_c N_p)} + W(n_s, n_c) \quad (4.10.19)$$

is a superposition of two-dimensional sinusoids in noise with frequencies $(\omega_k, N_p \omega_k)$ embedded in noise $W(n_s, n_c) = w(n_s + n_c N_p)$. Hence each 1D sinusoid in (4.10.17) with frequency ω is transformed into a 2D sinusoid with 2D-frequency $(\omega, N_p \omega)$. We express this relationship as

$$(\omega_{2D}, \check{\omega}_{2D}) = (\omega_{1D}, N_p \cdot \omega_{1D}). \quad (4.10.20)$$

As described in Section 4.5, the 2D-APES and 2D-Capon methods choose a set of snapshot vectors $\{\mathbf{z}_{\check{n}_s, \check{n}_c}\}$ whose elements are taken from consecutive sub-matrices of $X(n_s, n_c)$:

$$\mathbf{z}_{\check{n}_s, \check{n}_c} = \text{vec} \left(\begin{bmatrix} X(\check{n}_s, \check{n}_c) & \cdots & X(\check{n}_s, \check{n}_c + \check{N}_c - 1) \\ \vdots & \ddots & \vdots \\ X(\check{n}_s + \check{N}_s - 1, \check{n}_c) & \cdots & X(\check{n}_s + \check{N}_s - 1, \check{n}_c + \check{N}_c - 1) \end{bmatrix} \right) \quad (4.10.21)$$

for $\check{n}_s = 0, \dots, N_s - \check{N}_s$ and $\check{n}_c = 0, \dots, N_c - \check{N}_c$. It follows from (4.10.18) that the snapshots formed in (4.10.16) coincide with those formed in (4.10.21): $\mathbf{z}_{\check{n}_s, \check{n}_c} =$

$\mathbf{y}_{\check{n}_s + \check{n}_c N_p}$ and this fact establishes the sought relation between 2D-Capon/2D-APES and PG-Capon/PG-APES. As an aside, note that $W(n, \check{n})$ is a stationary random process provided that $w(n)$ is stationary (see [52] for a proof) and hence it makes sense to apply 2D-Capon or 2D-APES to $X(n_s, n_c)$.

To summarize, the PG-Capon and PG-APES methods can be formulated as follows:

1. Stack the available data segments onto each other to obtain $X(n_s, n_c)$ (see (4.10.18)).
2. Apply 2D Capon or APES in a standard fashion to the transformed data $X(n, \check{n})$, using a matrix filter of size $\check{N}_s \times \check{N}_c$. This way we obtain an estimate of the 2D amplitude spectrum $X(\omega, \check{\omega})$ corresponding to $X(n_s, n_c)$. Let this amplitude spectrum estimate be denoted by $\hat{\alpha}_{2D}(\omega, \check{\omega})$.
3. Obtain the spectral estimate of $x(n)$ by “inverting” the relation (4.10.20):

$$\hat{\alpha}(\omega) = \hat{\alpha}_{2D}(\omega, N_p \omega). \quad (4.10.22)$$

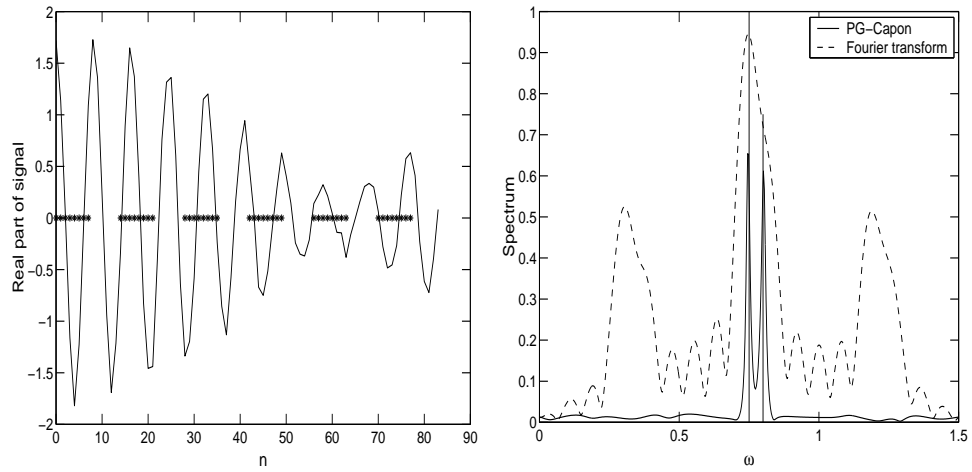
Based on the interpretation described above, we see that a fast implementation of PG-Capon and PG-APES can be obtained by using the techniques in Section 4.8 (with the only difference that it is sufficient to evaluate the spectral estimate for the frequencies $(\omega, N_p \cdot \omega)$ – and this can be accomplished via a standard 1D FFT).

Numerical Examples

Example 4.12: Sinusoidal spectrum and periodically gapped data.

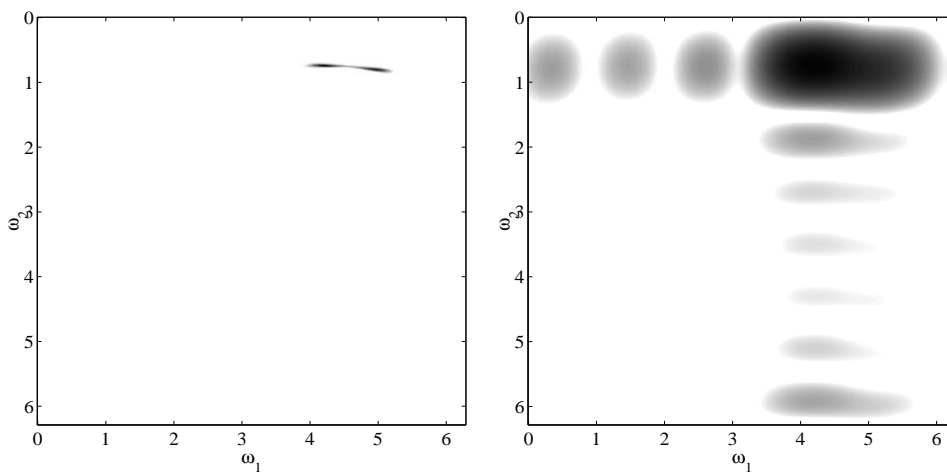
In this example (see also [52]) we consider a time-series $x(n)$ consisting of two complex sinusoids with amplitudes 1 and 0.75 and frequencies 0.75 and 0.8, respectively, embedded in white Gaussian noise with variance 0.01. We assume that $N_c = 6$ clusters with $N_s = 8$ samples each are available, and that the spacing between the clusters is $N_p = 14$. In Figure 4.16(a) we show the real part of the signal under consideration and indicate which samples that are available. In Figure 4.16(b), the PG-Capon spectral estimate (using $\check{N}_c = 3$ and $\check{N}_s = 5$) is shown together with the DFT spectral estimate. Note the artifacts in the Fourier spectrum and the superior resolution of PG-Capon compared with the Fourier transform.

In Figure 4.16(c) we illustrate the interpretation of PG-Capon discussed in Subsection 4.10.1. The figure shows the 2D spectral estimate of the fictitious 2D time-series using Capon, and Figure 4.16(d) shows the corresponding Fourier transform spectrum. We can think of the “horizontal dimension” as a “fast frequency axis” and the “vertical dimension” as a “slow frequency axis”. The horizontal dimension is needed to resolve closely spaced peaks, and the vertical dimension is needed to eliminate ambiguities. The resolution in the horizontal dimension is inversely proportional to the “aperture” (or data) length $N_c N_p$, and the resolution in the vertical dimension is inversely proportional to the number of samples per cluster N_s . ■



(a) Real part of signal (the stars indicate the available samples)

(b) DFT and PG-Capon spectral estimates (the solid vertical lines indicate the true amplitude spectrum)



(c) 2D spectral estimate of transformed time-series (4.10.18) obtained via Capon

(d) 2D spectral estimate of transformed time-series (4.10.18) obtained via the DFT

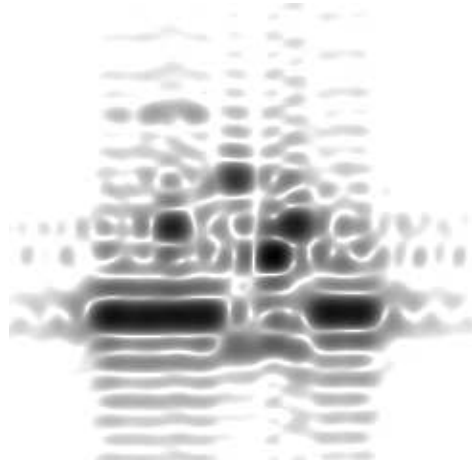
Figure 4.16. Results for Example 12.

Example 4.13: SAR imaging of an object.

We illustrate how PG-Capon can be applied to SAR imaging with angular diversity (see also [52]). We use a data matrix of size 50×50 generated by an electromagnetic prediction code [45], and we assume that the columns 6–10, 16–20, 26–30, 36–40, 46–50 are missing due to the angular diversity. Each row of the phase-history matrix is processed separately (this is clearly suboptimal, cf. Example 4.9, but a 2D version of PG-Capon is not yet available). Figure 4.17(a) shows a photograph of the object under consideration. In Figures 4.17(b) and 4.17(c) we show the Fourier and Capon images of the full phase history data matrix. Finally, Figures 4.17(d) and 4.17(e) show the Fourier and PG-Capon images of the gapped phase-history data. ■



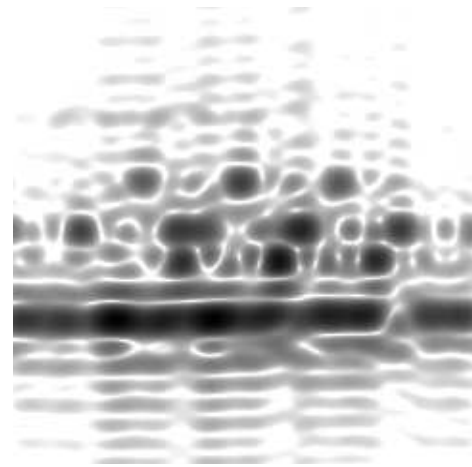
(a) Photograph of object



(b) 2D DFT image of complete data.



(c) 2D Capon image of complete data



(d) 2D Fourier image of gapped data matrix

Figure 4.17. (continued on the next page)



(e) PG-Capon image of gapped data matrix

Figure 4.17. (cont.) Results for Example 13. Dynamic range in all figures: 35dB (amplitude).

4.10.2 APES for Gapped Data: GAPES

The papers [53], [54], [36] developed an extension of APES to gapped data that (unlike the methods in the previous Subsection 4.10.1) can deal with quite general sampling patterns. The method proposed therein is called “gapped-data” APES (GAPES) and is based on interpolating the missing samples by minimizing the WLS criterion (4.3.9) with respect to the missing data as well. Despite the relationship between the Capon estimator and APES, a “GAPES-like” extension of Capon to gapped data appears to be unfeasible).

In this section, we describe the main ideas of GAPES. We treat only the 2D case, the 1D case follows as a special instance. As before, denote the $N \times \bar{N}$ data matrix by \mathbf{X} . We require $N \geq 1$ and $\bar{N} > 1$, and therefore the special case of $N = 1$ corresponds to the 1D case. Let \mathcal{U} be the set of sample indices (n, \bar{n}) for which data are unavailable (or missing), and let \mathcal{A} be the (complementary) set of sample indices (n, \bar{n}) for which data are available. The set of unavailable samples $\{X(n, \bar{n}) : (n, \bar{n}) \in \mathcal{U}\}$ is denoted by \mathcal{X}_u , whereas the set of available samples $\{X(n, \bar{n}) : (n, \bar{n}) \in \mathcal{A}\}$ is denoted by \mathcal{X}_a . Hence the problem of interest is to estimate $\alpha(\omega, \bar{\omega})$ given \mathcal{X}_a .

The GAPES Criterion

Following [36], we minimize (4.3.9) with respect to both the spectrum and the missing data using the filterbank framework. Consider the FB-APES criterion in (4.6.12), and assume that we want to compute $\alpha(\omega, \bar{\omega})$ and $\mathbf{H}(\omega, \bar{\omega})$ simultaneously for a set of $K \times \bar{K}$ frequencies $\{\omega_k, \bar{\omega}_{\bar{k}}\}_{k=0, \bar{k}=0}^{K-1, \bar{K}-1}$ that lie on a uniform grid; hence $\omega_k = 2\pi k/K$ and $\bar{\omega}_{\bar{k}} = 2\pi \bar{k}/\bar{K}$. The computation of $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ and $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ then amounts to solving $K\bar{K}$ LS minimization problems of the form (4.6.12) (one for each frequency $\omega_k, \bar{\omega}_{\bar{k}}$). It is clear that the solutions to these LS problems are the solutions to the following joint minimization problem:

$$\begin{aligned} & \min_{\{\alpha(\omega_k, \bar{\omega}_{\bar{k}}), \beta(\omega_k, \bar{\omega}_{\bar{k}}), \mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}} \\ & \sum_{k=0}^{K-1} \sum_{\bar{k}=0}^{\bar{K}-1} \left\{ \left\| \text{vec}(\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}}) * \mathbf{X}) - \alpha(\omega_k, \bar{\omega}_{\bar{k}}) \mathbf{a}_{L, \bar{L}}(\omega_k, \bar{\omega}_{\bar{k}}) \right\|^2 \right. \\ & \left. + \left\| \text{vec}(\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}}) * \tilde{\mathbf{X}}) - \beta(\omega_k, \bar{\omega}_{\bar{k}}) \mathbf{a}_{L, \bar{L}}(\omega_k, \bar{\omega}_{\bar{k}}) \right\|^2 \right\} \end{aligned} \quad (4.10.23)$$

subject to (4.6.11) and (4.6.13). Making use of (4.5.15), we can rewrite the APES criterion (4.10.23) as

$$\begin{aligned} & \min_{\{\alpha(\omega_k, \bar{\omega}_{\bar{k}}), \beta(\omega_k, \bar{\omega}_{\bar{k}}), \mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}} \\ & \left\{ \left\| \begin{bmatrix} \mathbf{T}\mathbf{G}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{T}\mathbf{G}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \text{vec}(\mathbf{X}) - \begin{bmatrix} \alpha(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \alpha(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \right\|^2 \right. \\ & \left. + \left\| \begin{bmatrix} \mathbf{T}\mathbf{G}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{T}\mathbf{G}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \text{vec}(\tilde{\mathbf{X}}) - \begin{bmatrix} \beta(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \beta(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \right\|^2 \right\} \end{aligned} \quad (4.10.24)$$

subject to (4.6.11) and (4.6.13) where \mathbf{T} is defined in (4.5.13) and the relation between $\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})$ and $\mathbf{G}(\omega_k, \bar{\omega}_{\bar{k}})$ is described by (4.5.14).

The GAPES algorithm obtains $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ by minimizing the criterion in

(4.10.24) with respect to the missing data \mathcal{X}_u as well:

$$\min_{\{\alpha(\omega_k, \bar{\omega}_{\bar{k}}), \beta(\omega_k, \bar{\omega}_{\bar{k}}), \mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}, \mathcal{X}_u} \left\{ \left\| \begin{bmatrix} \mathbf{TG}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{TG}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \text{vec}(\mathbf{X}) - \begin{bmatrix} \alpha(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \alpha(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{TG}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{TG}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \text{vec}(\tilde{\mathbf{X}}) - \begin{bmatrix} \beta(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \beta(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \right\|^2 \right\} \quad (4.10.25)$$

subject to (4.6.11) and (4.6.13). In (4.10.25) the matrix \mathbf{X} is of dimension $N \times \bar{N}$, although some of its elements are *unknown* parameters in the minimization. For (4.10.25) to be meaningful in practice, K and \bar{K} have to be fairly large.

The minimization of (4.10.25) with respect to \mathcal{X}_u has a clear intuitive interpretation in terms of *interpolating* the missing data. The missing samples \mathcal{X}_u are interpolated under the only, but natural, assumption that their amplitude spectrum is similar to that of the available data \mathcal{X}_a . In some applications, the interpolated data may be of interest *per se*.

The article [36] proposes to use a cyclic minimization to find the minimizer (4.10.25). Such a minimization is carried out via the following steps:

- (a) Obtain an initial spectrum estimate $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ along with the corresponding filters $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$.
- (b) Minimize the criterion in (4.10.25) with respect to \mathcal{X}_u (keeping $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ and $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ fixed).
- (c) Minimize the criterion in (4.10.25) with respect to $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ and $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ keeping \mathcal{X}_u fixed, in other words re-compute the estimated spectrum given \mathcal{X}_u from Step (b).
- (d) Iterate Steps (b)-(c) until convergence.

Step (c) is nothing more than the computation of the APES spectrum. Before discussing the other two steps in detail, let us note that the criterion in (4.10.25) is non-negative and non-increasing in both Step (b) and Step (c); hence the algorithm will converge. Since the criterion in (4.10.25) is non-convex, there is no guarantee that the point of convergence is the global minimum (this is the usual case for non-convex optimization problems). Empirical experience reported in [36] indicates that with a proper initialization in Step (a), it is possible to come close to the global minimum using a few iterations.

Step (a): Obtaining an initial spectral estimate Initial estimates of the spectrum $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ and the filters $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ can be obtained by choosing an initial size $M_0 \times \bar{M}_0$ of the filter matrix and thereafter determining the spectrum estimates $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ and the filters $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ by minimizing the APES criterion (4.10.24) *using the available data only*. In other words, the vector inside the norm in the criterion in (4.10.24) is modified such that to remove all elements which involve \mathcal{X}_u .

Let \mathcal{S} be the set of sample indices (l, \bar{l}) such that

$$\{(l, \bar{l}), \dots, (l, \bar{l} + \bar{M}_0 - 1), \dots, (l + M_0 - 1, \bar{l}), \dots, (l + M_0 - 1, \bar{l} + \bar{M}_0 - 1)\} \in \mathcal{A}. \quad (4.10.26)$$

Define the set of $M_0 \bar{M}_0 \times 1$ vectors $\{\mathbf{y}_{l, \bar{l}}^0 : (l, \bar{l}) \in \mathcal{S}\}$ (which contain available data samples only) similarly to (4.5.2), and let $|\mathcal{S}|$ be the number of elements in \mathcal{S} . Furthermore, define the initial forward sample covariance matrix

$$\hat{\mathbf{R}}_f^0 \triangleq \frac{1}{|\mathcal{S}|} \sum_{(l, \bar{l}) \in \mathcal{S}} \sum_{(l, \bar{l}) \in \mathcal{S}} \mathbf{y}_{l, \bar{l}}^0 \mathbf{y}_{l, \bar{l}}^{0H}. \quad (4.10.27)$$

Next, define the initial backward and forward-backward sample covariance matrices $\hat{\mathbf{R}}_b^0$ and $\hat{\mathbf{R}}_{fb}^0$, the Fourier transforms $\bar{\mathbf{g}}^0(\omega, \bar{\omega})$, $\tilde{\mathbf{g}}^0(\omega, \bar{\omega})$ and the matrix $\hat{\mathbf{Q}}^0(\omega, \bar{\omega})$ (cf. (4.6.4), (4.6.5), (4.6.6) and (4.6.7)). The initial estimates are now given by (4.6.9) but with $\bar{\mathbf{g}}(\omega, \bar{\omega})$ and $\hat{\mathbf{Q}}(\omega, \bar{\omega})$ replaced with $\bar{\mathbf{g}}^0(\omega, \bar{\omega})$ and $\hat{\mathbf{Q}}^0(\omega, \bar{\omega})$, respectively. Note that all quantities denoted by $(\cdot)^0$ in this paragraph are functions of the available data only; no interpolation of the missing data has taken place yet. Once the interpolation Step (b) has been carried out, the corresponding quantities can be built from the full (interpolated) data matrix and consequently the ‘‘conventional’’ APES algorithm is used in Step (c).

The size of the initial filter matrix $M_0 \times \bar{M}_0$ must be chosen such that $\hat{\mathbf{R}}_{fb}^0$ has full rank. Since the spectral resolution typically increases with increasing filter lengths, it is argued in [36] that M_0 and \bar{M}_0 should be chosen equal to the largest integers that render $\hat{\mathbf{R}}_{fb}^0$ nonsingular. In the event that there are no such integers, some other initialization technique must be used; for instance, simply replacing all the missing samples with zeros and computing the initial spectrum estimate by applying the APES algorithm to the so-obtained sequence.

Step (b): Minimization of (4.10.25) with respect to \mathcal{X}_u It can be shown [11] that the filters $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ that minimize (4.10.23) (or, equivalently, (4.10.24)) satisfy the following conjugate symmetry constraint:

$$\begin{aligned} \mathbf{H}_{m, \bar{m}}(\omega, \bar{\omega}) &= \mathbf{H}_{M-1-m, \bar{M}-1-\bar{m}}^*(\omega, \bar{\omega}) \cdot e^{-i((M-1)\omega + (\bar{M}-1)\bar{\omega})} \\ &\triangleq \tilde{\mathbf{H}}_{m, \bar{m}}(\omega, \bar{\omega}) e^{i((M-1)\omega + (\bar{M}-1)\bar{\omega})} \end{aligned} \quad (4.10.28)$$

Making use of this property together with (4.6.13), it follows that the two terms in the sum in (4.10.25) are equal for all \mathbf{X} and $\tilde{\mathbf{X}}$ that obey (4.6.1). This is so since

for any $\omega, \bar{\omega}$ it holds that (see [36] for a proof):

$$\begin{aligned} & \|\mathbf{TG}(\omega, \bar{\omega}) \text{vec}(\mathbf{X}) - \alpha(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}(\omega, \bar{\omega})\| \\ &= \|\mathbf{TG}(\omega, \bar{\omega}) \text{vec}(\tilde{\mathbf{X}}) - \beta(\omega, \bar{\omega}) \mathbf{a}_{L, \bar{L}}(\omega, \bar{\omega})\|. \end{aligned} \quad (4.10.29)$$

Hence, since $\{\mathbf{H}(\omega_k, \bar{\omega}_{\bar{k}})\}$ and $\{\alpha(\omega_k, \bar{\omega}_{\bar{k}})\}$ are kept fixed during the minimization of (4.10.25) with respect to \mathcal{X}_u , Step (b) amounts to:

$$\min_{\mathcal{X}_u} \left\| \begin{bmatrix} \mathbf{TG}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{TG}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \text{vec}(\mathbf{X}) - \begin{bmatrix} \alpha(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \alpha(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix} \right\|^2. \quad (4.10.30)$$

To solve (4.10.30), let the matrices $\mathbf{G}_a(\omega_k, \bar{\omega}_{\bar{k}})$ and $\mathbf{G}_u(\omega_k, \bar{\omega}_{\bar{k}})$ be defined implicitly by the following equality:

$$\mathbf{G}(\omega_k, \bar{\omega}_{\bar{k}}) \text{vec}(\mathbf{X}) = \mathbf{G}_a(\omega_k, \bar{\omega}_{\bar{k}}) \text{vec}(\mathbf{X}_a) + \mathbf{G}_u(\omega_k, \bar{\omega}_{\bar{k}}) \text{vec}(\mathbf{X}_u) \quad (4.10.31)$$

where \mathbf{X}_a and \mathbf{X}_u are matrices that contain the elements of \mathcal{X}_a and \mathcal{X}_u , respectively. In other words, $\mathbf{G}_a(\omega_k, \bar{\omega}_{\bar{k}})$ and $\mathbf{G}_u(\omega_k, \bar{\omega}_{\bar{k}})$ contain the columns of $\mathbf{G}(\omega_k, \bar{\omega}_{\bar{k}})$ that correspond to the indices in \mathcal{A} and \mathcal{U} , respectively. Furthermore, introduce the following matrices:

$$\begin{aligned} \check{\mathbf{G}}_a &\triangleq \begin{bmatrix} \mathbf{TG}_a(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{TG}_a(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix}, \\ \check{\mathbf{G}}_u &\triangleq \begin{bmatrix} \mathbf{TG}_u(\omega_0, \bar{\omega}_0) \\ \vdots \\ \mathbf{TG}_u(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix}. \end{aligned} \quad (4.10.32)$$

Then the criterion (4.10.30) can be written as

$$\min_{\mathbf{X}_u} \left\| \check{\mathbf{G}}_u \text{vec}(\mathbf{X}_u) + \check{\mathbf{G}}_a \text{vec}(\mathbf{X}_a) - \alpha \right\|^2 \quad (4.10.33)$$

where

$$\alpha \triangleq \begin{bmatrix} \alpha(\omega_0, \bar{\omega}_0) \mathbf{a}_{L, \bar{L}}(\omega_0, \bar{\omega}_0) \\ \vdots \\ \alpha(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \mathbf{a}_{L, \bar{L}}(\omega_{K-1}, \bar{\omega}_{\bar{K}-1}) \end{bmatrix}. \quad (4.10.34)$$

Since (4.10.33) is a standard LS problem, a closed form solution is easily obtained:

$$\text{vec}(\hat{\mathbf{X}}_u) = (\check{\mathbf{G}}_u^H \check{\mathbf{G}}_u)^{-1} \check{\mathbf{G}}_u^H (\alpha - \check{\mathbf{G}}_a \text{vec}(\mathbf{X}_a)). \quad (4.10.35)$$

Numerically, the solution to (4.10.33) can be obtained by QR-factorization (see, e.g., [35], [55]). However, there are at least two drawbacks with this. First, the QR factorization requires the storage of $\check{\mathbf{G}}_u$ which is approximately of dimension $L\bar{L}K\bar{K} \times |\mathcal{U}|$. This can be virtually impossible even for data records of moderate size. Second, the QR factorization requires $O(L\bar{L}K\bar{K}|\mathcal{U}|^2)$ floating point operations [35]. Since the equation system corresponding to (4.10.35) is highly structured, it can be solved more efficiently than by using the QR factorization. In [36] it was proposed to solve (4.10.35) by using the conjugate gradient (CG) algorithm [35].

The CG algorithm is an iterative procedure, and the number of required iterations in the CG algorithm is bounded by the number of unknowns. Each iteration in the CG algorithm requires the multiplication of $\check{\mathbf{G}}_u^H \check{\mathbf{G}}_u$ with a vector. By exploiting the Toeplitz-like structure of \mathbf{G}_u and using an approach similar to that in Section 4.C, this multiplication can be done in $O(K\bar{K}N\bar{N} \log(N\bar{N}))$ operations (see Subsection 4.5.1). Hence the total cost of minimizing (4.10.33) is at most $O(K\bar{K}N\bar{N}|\mathcal{U}| \log(N\bar{N}))$ operations, which is typically much less than the complexity of a QR-factorization. Also, if (4.10.33) is minimized via the CG algorithm, the storage of the regressor matrix $\check{\mathbf{G}}_u$ is avoided (which would require $O(N\bar{N}K\bar{K}|\mathcal{U}|)$ bits of memory).

Numerical Examples

We next present two examples of application of the GAPES algorithm. Note that for large data matrices, the direct application of the 2-D GAPES algorithm can be computationally demanding, even if an efficient technique is employed to solve (4.10.33). In the first example, we present a technique to cope with large data matrices. This example also illustrates the advantage of using 2D interpolation instead of 1D data processing.

Example 4.14: GAPES applied to line spectrum.

In this example (see also [36]) we consider a data matrix of size 32×50 consisting of three noisy sinusoids, with the same frequencies, amplitudes and noise variance as in Example 4.8. All samples in the columns 10–20 and 30–40 are missing. The true spectrum is shown in Figure 4.18(a) and the missing-data pattern is shown in Figure 4.18(b). In Figure 4.18(c) we show the 512×512 windowed Fourier transform (WFT) spectrum of the full data obtained using a 2D Taylor window with shape parameter 5 and sidelobe level -35 dB. In Figure 4.18(d) we show the 512×512 APES spectrum of the full data obtained by using a filter matrix of size 8×8 . Figure 4.18(e) shows the WFT spectrum obtained by setting the missing samples to zero.

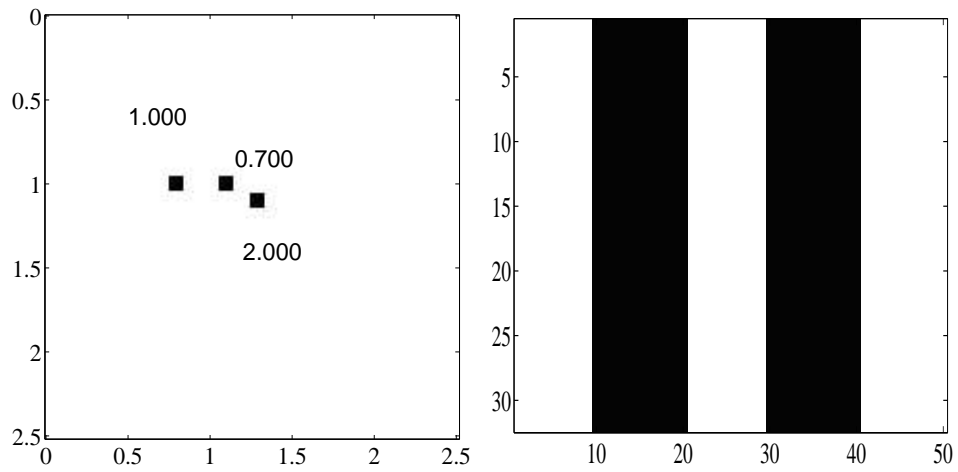
Next we apply the following two methods:

- (a) *1D (row-wise) data processing.* Here each row in the data matrix is interpolated separately. This is done by applying the 1D-GAPES algorithm with $N = 1$ and $\bar{N} = 50$ using a 1D filter of length $\bar{M} = 4$ and a frequency grid with $\bar{K} = 128$ points. After completing this interpolation for all rows of the data matrix, we

obtain the final spectral estimate by using the 2D-APES algorithm. The results after one iteration are shown in Figure 4.18(f) (further iteration did not change the result visibly).

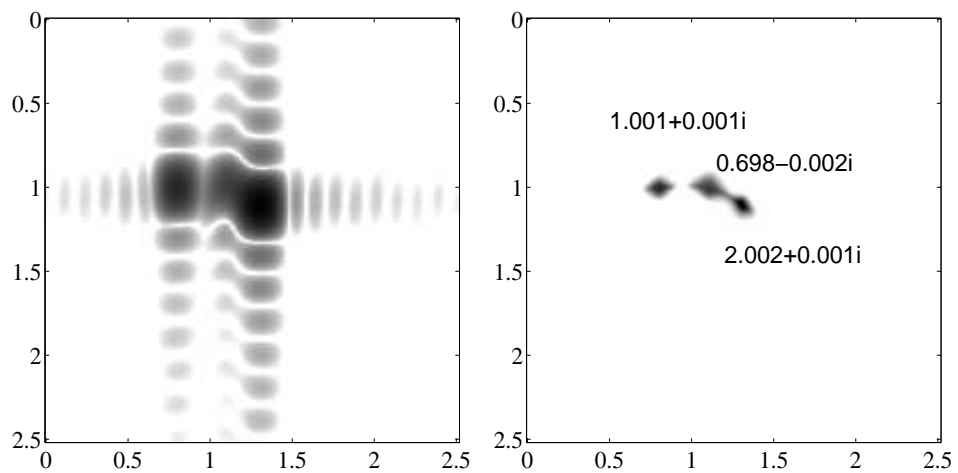
- (b) *Partial 2D data processing.* Here, we divide the data matrix into 4 blocks of size 8×50 and *interpolate the missing data in each block separately*. To perform the interpolation, we apply 2D-GAPES with a filter matrix of size 4×4 and a frequency grid of size 16×128 . After the interpolation step is completed, the spectrum of the so-obtained interpolated data matrix is computed via 2D-APES. The result after one iteration is shown in Figure 4.18(g) (further iteration did not change the result visibly).

We compare the results with that of the CLEAN method (used as a deconvolution method with a grid size of 1024×1024 , a loop gain of 0.5 and iterating 200 times, although the result was not very sensitive to these parameters). In Figure 4.18(h) we show the WFT image of the data reconstructed by CLEAN, and in Figure 4.18(i) we show the APES spectrum estimate of the data matrix obtained by CLEAN. It is clear that GAPES outperforms the methods based on CLEAN. We also conclude that the gain of using (even partial) 2D data interpolation can be significant. ■



(a) True amplitude spectrum

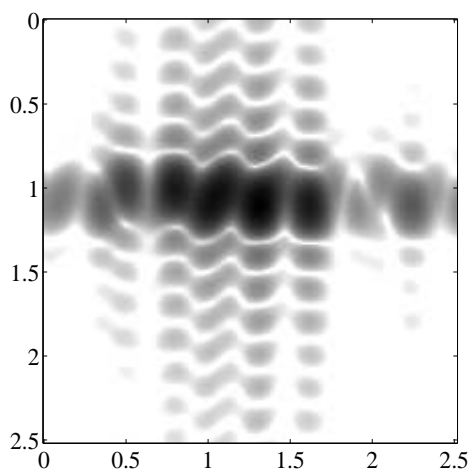
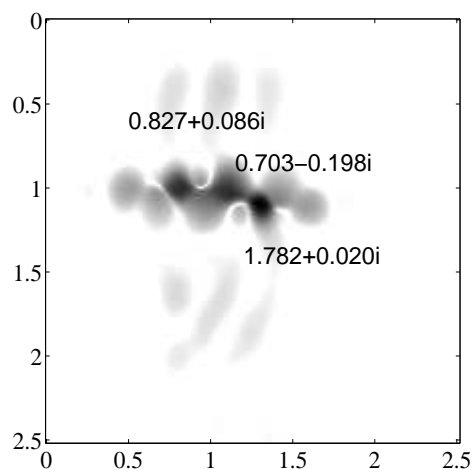
(b) Missing data pattern (black areas correspond to missing data)



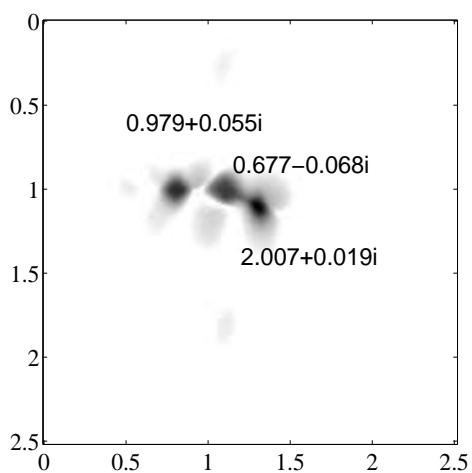
(c) WFT of full data

(d) APES of full data

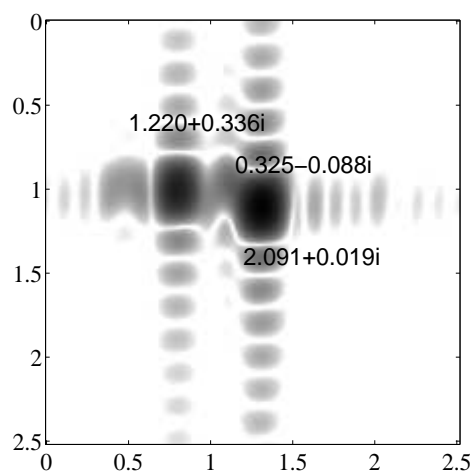
Figure 4.18. (continued on the next page)

(e) WFT with $\mathcal{X}_u = \mathbf{0}$ 

(f) GAPES spectrum after one iteration with 1D-interpolation of one row at a time

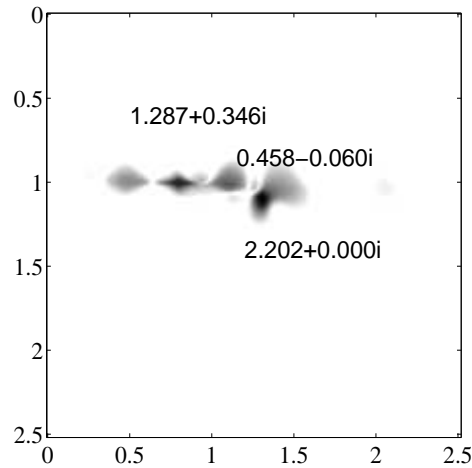


(g) GAPES spectrum after one iteration with 2D-interpolation of 8 rows at a time



(h) WFT of data matrix reconstructed via CLEAN

Figure 4.18. (continued on the next page)



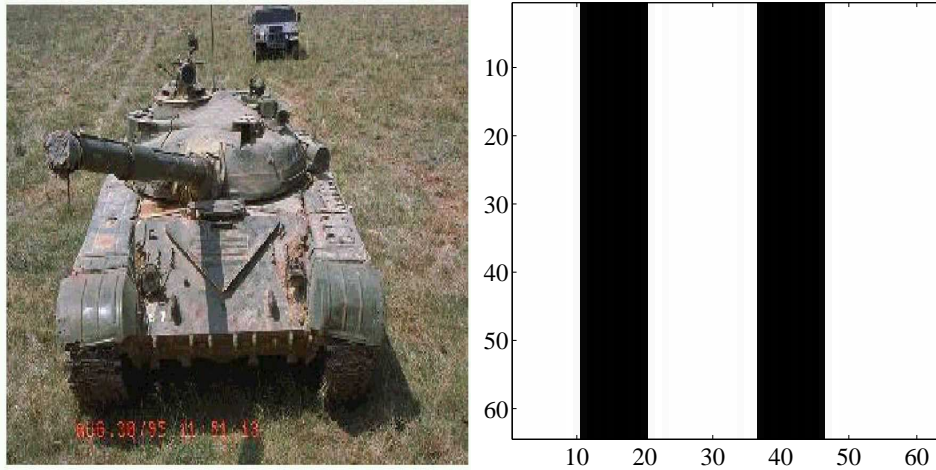
(i) APES of data matrix reconstructed via CLEAN

Figure 4.18. (cont.) Results for Example 14.

Example 4.15: GAPES applied to measured SAR data.

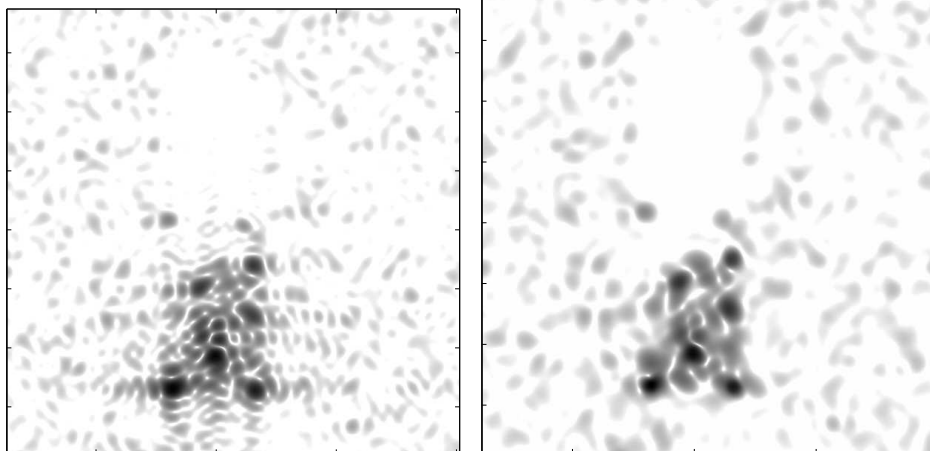
In this example (see also [56]) we apply the GAPES algorithm to *measured* SAR data. We consider a 64×64 phase history matrix collected by the Sandia National Laboratory using the STARLOS sensor. The radar system operates in spotlight mode and uses a center frequency of 9.6 GHz and a bandwidth of 590 MHz. The object illuminated by the radar is a tank (depicted in Figure 4.19(a)) and is located at a distance of 4.5km and an elevation angle of 15° from the radar. To simulate the angular diversity, we create artificial gaps in the phase history matrix by removing the columns (corresponding to cross-range samples) 11–20 and 37–46, as illustrated in Figure 4.19(b). In Figure 4.19(c) we show the SAR image obtained by applying WFT (as described above) to the full data. Figure 4.19(d) shows the image obtained by the application of APES to the full data with $M = \bar{M} = 16$ (note the superior resolution of APES compared with that of WFT). In Figure 4.19(e) we show the result of applying WFT to the phase history data with the missing data \mathbf{X}_u set to zero (note the artifacts in the image due to the gapping). Figures 4.19(f) and 4.19(g) show the results of applying GAPES with 1D (row-by-row) interpolation and partial 2D (16 rows at a time) processing, respectively. We used the same filter lengths as for Example 4.14. It is clear also from this example that by applying GAPES to the incomplete phase history data we obtained a SAR image which is almost identical to the 2D-APES image in Figure 4.19(d) (especially note that the

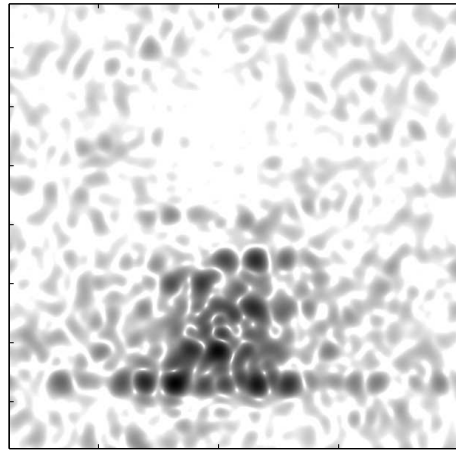
“shadow”, which is characteristic of SAR images, remains essentially unchanged in the 2D-GAPES image). ■



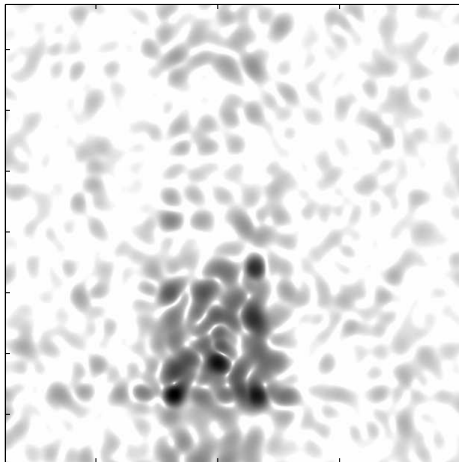
(a) Photograph of the object

(b) Topology of data matrix (black areas correspond to missing data)

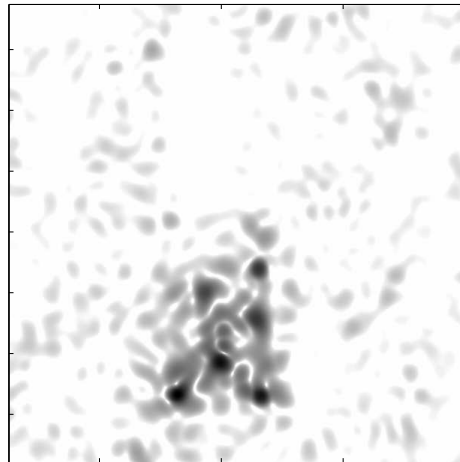
(c) SAR image of the object (at 0° azimuth angle) obtained via WFT of full data(d) SAR image of the object (at 0° azimuth angle) obtained via APES of full data with $M = \bar{M} = 16$ **Figure 4.19.** (continued on the next page)



(e) SAR image of the object (at 0° azimuth angle) obtained via WFT with $\mathbf{X}_u = \mathbf{0}$



(f) SAR image of the object (at 0° azimuth angle) obtained via GAPES after two iterations with 1D-interpolation of one row at a time



(g) SAR image of the object (at 0° azimuth angle) obtained via GAPES after two iterations with 2D-interpolation of 16 rows at a time

Figure 4.19. (cont.) Results for Example 4.15.

4.11 Special Topics

Spectral estimation via adaptive filterbank methods is a theme that can be extended and developed in many ways. The intention of this section is to summarize some extensions and variations of the filterbank methods that have appeared in the literature. Instead of providing a full treatment of all techniques, we refer the reader to the original articles for more details.

4.11.1 APES and Capon for Real-Valued Data

The estimation problem formulated in Section 4.3 as well as the associated solution are meaningful in a strict sense only when the data fulfill a model such as that in (4.3.1). An important special case of (4.3.1) is when the observed data are *real-valued*, and can be modelled as a superposition of real sinusoids embedded in noise. In this case, we can decompose the signal in a sinusoidal term with amplitude $\alpha(\omega)$ and phase $\phi(\omega)$ and a residual term:

$$\begin{aligned} x(n) &= \alpha(\omega) \cos(\omega n + \phi(\omega)) + w_\omega(n) \\ &= \alpha_I(\omega) \cos(\omega n) + \alpha_Q(\omega) \sin(\omega n) + w_\omega(n) \end{aligned} \quad (4.11.1)$$

where $\alpha_I(\omega), \alpha_Q(\omega)$ can be interpreted as in-phase and quadrature components of the sinusoid $\sin(\omega n + \phi(\omega))$, respectively. As before, the decomposition in (4.11.1) is meaningful only for a fixed ω of interest.

Nonparametric spectral analysis of real-valued data was studied in [57] by taking a filterbank approach (similar to that in Section 4.4). It was suggested to use the same Capon and APES filter criteria as in (4.4.7) and (4.4.11), but with the constraint

$$\begin{aligned} \mathbf{a}^H(\omega)\mathbf{h}(\omega) &= 1 \\ \mathbf{a}^T(\omega)\mathbf{h}(\omega) &= 1 \end{aligned} \quad (4.11.2)$$

in lieu of (4.4.8). Obviously, the constraint in (4.11.2) is equivalent to requiring that the filter transfer function has a value of unity at *both* ω and $-\omega$. This is a natural constraint in view of the fact that real-valued signals have symmetric spectra.

Let

$$\mathbf{A}(\omega) = \begin{bmatrix} 1 & 0 \\ \cos(\omega) & \sin(\omega) \\ \vdots & \vdots \\ \cos((M-1)\omega) & \sin((M-1)\omega) \end{bmatrix}. \quad (4.11.3)$$

Then [57] showed that the resulting Capon and APES spectral estimate can be written:

$$\begin{bmatrix} \hat{\alpha}_I(\omega) \\ \hat{\alpha}_Q(\omega) \end{bmatrix} = \hat{\alpha}(\omega) = \mathbf{G}_\omega^{-1} \mathbf{C}_\omega^T \mathbf{\Phi}^{-1}(\omega) \mathbf{A}(\omega) (\mathbf{A}^T(\omega) \mathbf{\Phi}^{-1}(\omega) \mathbf{A}(\omega))^{-1} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4.11.4)$$

where

$$\begin{aligned}\mathbf{G}_\omega &= \frac{1}{L} \sum_{l=0}^{L-1} \begin{bmatrix} \cos(\omega l) & \sin(\omega l) \\ \sin(\omega l) & \cos(\omega l) \end{bmatrix} \\ \mathbf{C}_\omega &= \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}_l \cdot [\cos(\omega l) \quad \sin(\omega l)]\end{aligned}\tag{4.11.5}$$

In (4.11.4) and (4.11.5), $\{\mathbf{y}_l\}$ are defined as in Section 4.3, and $\Phi(\omega)$ is a weighting matrix chosen according to:

$$\begin{aligned}\Phi_{\text{Capon}}(\omega) &= \hat{\mathbf{R}} = \frac{1}{L} \mathbf{Y} \mathbf{Y}^T \\ \Phi_{\text{APES}}(\omega) &= \frac{1}{L} \mathbf{Y} \mathbf{Y}^T - \mathbf{C}_\omega \mathbf{G}_\omega^{-1} \mathbf{C}_\omega^T\end{aligned}\tag{4.11.6}$$

for Capon and APES, respectively.

The article [57] observed empirically that the estimator designed for real-valued sinusoids outperforms that designed for complex sinusoids, although the difference was not substantial. The same authors also proved analytically that the estimators designed for complex data will have a slight bias if the data are real-valued.

4.11.2 Combining APES and Capon: CAPES

We know that the choice of snapshot (or filter) length M is a way to trade resolution against statistical stability of the spectral estimates and that the resolution capability of Capon and APES, and their FB averaged variants, differ substantially in finite samples. In [27] (and Example 4.5) it was found that the Capon estimator (without FB averaging) is the method that gives the highest resolution. In particular, for a given M , Capon yields a better resolution than APES; consequently Capon may be a preferable choice for locating spectral lines. Furthermore, [27] observed that FB-APES usually gives superior estimates of the amplitude spectrum *at the true location of a spectral line*.

The conclusions of [27] led to an intuitively appealing and computationally tractable algorithm that combines the resolution properties of Capon with the spectral estimation accuracy of APES. The new algorithm was called Capon-APES (CAPES) and has a very simple structure. Capon is first used to locate the spectral lines, and APES is subsequently used to recover their amplitudes. The computational complexity of the resulting estimator is of the same order as that of Capon (the extra computations needed to evaluate the APES spectrum at the spectral line locations are negligible).

4.11.3 APES and Capon for Damped Sinusoids

A recent contribution [58] presents an extension of the Capon and APES methods to the case of *damped* sinusoids in noise. In this case there are two parameters associated with each sinusoid, namely an angular frequency ω and a damping coefficient

λ . The associated spectrum can be considered to be two-dimensional, with ω and λ as variables. The method in [58] was applied to synthetic data as well as real data from a nuclear magnetic resonance (NMR) application where the data can often be quite accurately modelled as a superposition of damped sinusoids in noise.

The results of [58] are easy to describe in the framework of this chapter. Let $x(n)$ be an arbitrary one-dimensional (1D) time series. Following the developments in Section 4.3, for a given frequency ω and a given damping coefficient λ , let us decompose $x(n)$ as:

$$x(n) = \alpha(\omega, \lambda)e^{i\omega n - \lambda n} + w_{\omega, \lambda}(n) \quad (4.11.7)$$

and assume that N consecutive samples of $x(n)$ are arranged in an $M \times L$ Hankel matrix \mathbf{Y} according to (4.3.2). Also let $\mathbf{A}(\omega)$ be the following rank-one matrix:

$$\mathbf{A}(\omega, \lambda) = \mathbf{a}_M(\omega, \lambda)\mathbf{a}_L^T(\omega, \lambda) \quad (4.11.8)$$

where

$$\mathbf{a}_P(\omega, \lambda) \triangleq \begin{bmatrix} 1 \\ e^{i\omega - \lambda} \\ \vdots \\ e^{i\omega(P-1) - \lambda(P-1)} \end{bmatrix} \quad (4.11.9)$$

for any integer P . Then we can write

$$\mathbf{Y} = \alpha(\omega, \lambda)\mathbf{A}(\omega, \lambda) + \mathbf{W}(\omega, \lambda). \quad (4.11.10)$$

The relatives of Capon and APES for damped sinusoids are easily formed by estimating $\alpha(\omega, \lambda)$ in (4.11.10) in a WLS sense:

$$\hat{\alpha}(\omega, \lambda) = \underset{\alpha(\omega, \lambda)}{\operatorname{argmin}} \left\| \Phi^{-1/2}(\omega, \lambda)(\mathbf{Y} - \alpha(\omega, \lambda)\mathbf{A}(\omega, \lambda)) \right\|^2. \quad (4.11.11)$$

The solution to this WLS problem takes on the same form as (4.3.10):

$$\hat{\alpha}(\omega, \lambda) = \frac{\mathbf{a}_M^H(\omega, \lambda)\Phi^{-1}(\omega, \lambda)\mathbf{g}(\omega, \lambda)}{\mathbf{a}_M^H(\omega, \lambda)\Phi^{-1}(\omega, \lambda)\mathbf{a}_M(\omega, \lambda)} \quad (4.11.12)$$

where

$$\mathbf{g}(\omega, \lambda) \triangleq \frac{1}{L(\lambda)}\mathbf{Y}\mathbf{a}_L^*(\omega, \lambda) \quad (4.11.13)$$

and

$$L(\lambda) = \sum_{n=0}^{L-1} e^{-2\lambda n} = \frac{e^{-2\lambda L} - 1}{e^{-2\lambda} - 1}. \quad (4.11.14)$$

The weight matrix $\Phi(\omega, \lambda)$ is formed in the same way as in Section 4.3, but with $\mathbf{a}_M(\omega)$ replaced by $\mathbf{a}_M(\omega, \lambda)$ and $\mathbf{g}(\omega, \lambda)$ taken as in (4.11.13) (see [58] for details).

There is a, somewhat superficial, similarity between the Capon/APES spectra for 1D damped sinusoids and the 2D spectra derived in Section 4.5: both are computed from the data via an equation of the form (4.11.12) depending on two variables. However, in the case of damped sinusoids, the vectors $\mathbf{a}_M(\omega, \lambda)$ are not “Fourier vectors” and thus the associated computations *cannot* be carried out directly via the techniques in Section 4.8.

4.11.4 Time-Recursive Implementations of Capon and APES

There are applications where it may be of interest to obtain an implementation of Capon and APES that is recursive in time. In words, given a sequence of N samples $\{x(0), \dots, x(N-1)\}$ along with the associated spectral estimate, the problem is to determine an updated spectral estimate corresponding to the sequence $\{x(1), \dots, x(N)\}$ where the first sample $x(0)$ is discarded and a new observation $x(N)$ is appended to the sequence.

Although no time-recursive implementation of the fast spectral estimators in [32] (see Section 4.8) is available, [59], [65] presented time-recursive variants of Capon and APES, respectively. The main idea is to apply the matrix inversion lemma to obtain an update of the inverse of the sample covariance matrix $\hat{\mathbf{R}}^{-1}$. Although the techniques in the cited papers are considerably faster than a brute-force implementation, it is an open question whether there remains additional structure in the problem that could be exploited by, for example, methods such as those in Section 4.8. An affirmative answer to this question could lead to a time-recursive implementation that would be fast and therefore quite useful in practice.

We will not dwell into the details of the time-recursive implementations in [59], [65], but we shall illustrate by an example how APES and Capon can be used for short-time spectral analysis tasks.

Example 4.16: Short-term spectral analysis of non-stationary 1D sequence.

We consider a data sequence consisting of four superimposed chirp signals:

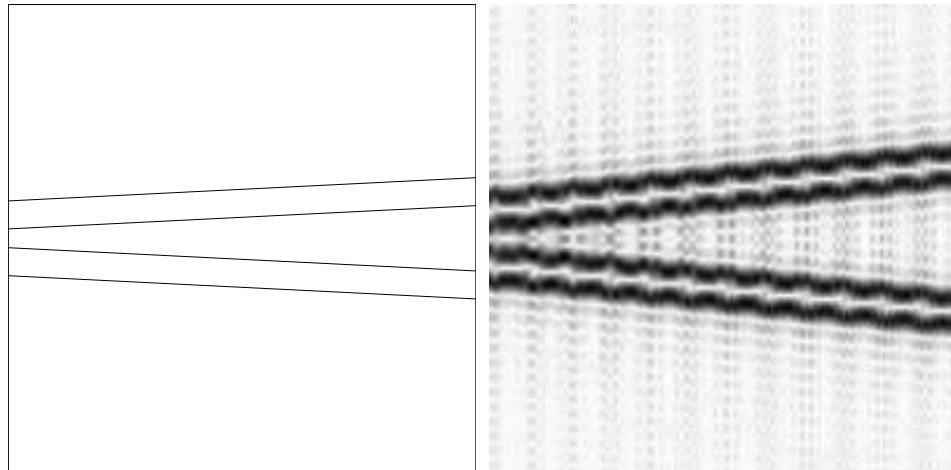
$$x(n) = \sum_{k=1}^4 e^{i2\pi f_k(n)n} + w(n) \quad (4.11.15)$$

for $n = 0, \dots, 231$ where

$$\begin{aligned} f_1(n) &= -0.08 - \frac{n}{4 \times 10^3} \\ f_2(n) &= 0.02 + \frac{n}{4 \times 10^3} \\ f_3(n) &= -0.02 - \frac{n}{4 \times 10^3} \\ f_4(n) &= 0.08 + \frac{n}{4 \times 10^3} \end{aligned} \tag{4.11.16}$$

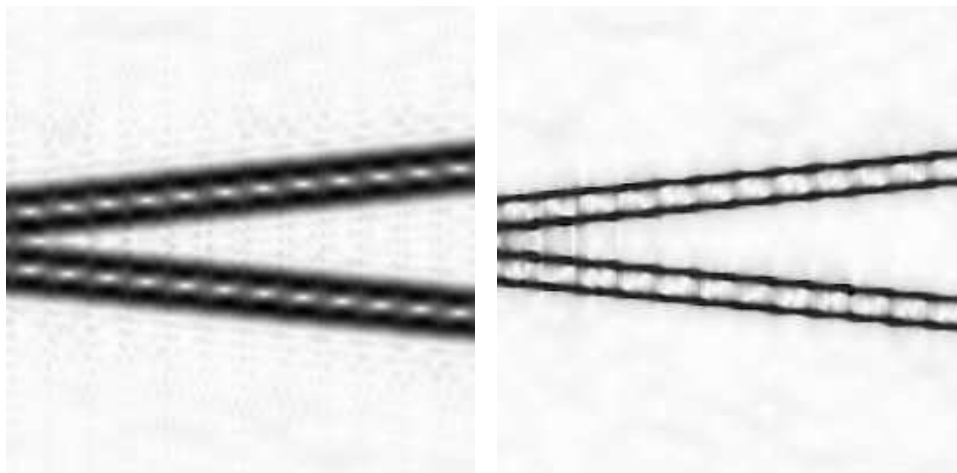
and w_n is zero-mean white complex Gaussian noise with variance = 0.01 (see also [65]).

Figure 4.20(a) shows the true amplitude spectrum of the signal as a function of time. In Figure 4.20(b) we show the spectral estimates obtained by using a 1D short-time Fourier transform (STFT). Figure 4.20(c) shows the estimates obtained by a 1D widowed STFT, and finally Figure 4.20(d) shows the result of applying 1D APES with maximally overlapping short-time data sequences of length $N = 32$. We can see that the APES method gives much better spectral estimates than the STFT and widowed STFT methods. ■



(a) True spectrum as a function of time

(b) STFT



(c) Windowed STFT

(d) APES with maximally overlapping short-time data sequences of length $N = 32$ **Figure 4.20.** Results for Example 4.16.

4.12 Conclusions

This chapter has summarized some main results in the area of nonparametric spectral analysis via adaptive filterbank methods. We started by formulating the spectral estimation problem as a weighted least-squares problem and showed how various nonparametric algorithms can be derived in a natural manner in this framework. Next we discussed how the nonparametric methods can be interpreted in the context of adaptive filterbanks, and showed how they can be computed efficiently. We also discussed various extensions of the methods, with the extension to gapped data as a particularly important case.

Although there exist an abundance of research papers in this area, there are still topics that should form the subject of future investigations. We will not give an exhaustive list of open problems, but in our opinion the most interesting and relevant ones to solve include (1) finding a fast time-recursive implementation of Capon and APES, (2) obtaining a fast implementation for the estimator in Subsection 4.11.3, and (3) deriving an “optimal” (in finite samples) weighting matrix for the MMSE spectral estimator.

Acknowledgements

Parts of this work were supported by the National Science Foundation grant no. CCR-0104887 and the Swedish Foundation for Strategic Research. The authors would like to thank Dr. Andreas Jakobsson, Yngve Selén, and Yanwei Wang for offering comments on parts of the manuscript.

APPENDICES

4.A Proof of Theorem 4.1

Since the data $x(n)$ is an outcome of (4.2.2), it holds that

$$\mathbf{y}_l = \sum_{k=1}^K \alpha_k e^{i\omega_k l} \mathbf{a}_M(\omega_k) + \mathbf{w}_l(\omega) \quad (4.A.1)$$

for $l = 0, \dots, L-1$ and hence it follows that

$$\hat{\alpha}(\omega) = \alpha(\omega) + \phi(\omega) + \psi(\omega) \quad (4.A.2)$$

where

$$\begin{aligned} \phi(\omega) &= \frac{1}{L} \sum_{k=1, k: \omega_k \neq \omega}^K \alpha_k \mathbf{h}^H(\omega) \mathbf{a}_M(\omega_k) \sum_{l=0}^{L-1} e^{i(\omega_k - \omega)l} \\ \psi(\omega) &= \frac{1}{L} \mathbf{h}^H(\omega) \sum_{l=0}^{L-1} \mathbf{w}_l(\omega) e^{-i\omega l} \end{aligned} \quad (4.A.3)$$

From (4.A.2) it follows that

$$E[|\hat{\alpha}(\omega) - \alpha(\omega)|^2] = E[|\phi(\omega)|^2] + 2 \Re \{ E[\phi^*(\omega) \psi(\omega)] \} + E[|\psi(\omega)|^2]. \quad (4.A.4)$$

In a usual fashion, let “ $f(x) = O(g(x))$ ” denote the existence of a constant C such that $|f(x)| \leq Cg(x)$ for all x . Since

$$\left| \sum_{l=0}^{L-1} e^{i\omega l} \right| = O(1) \quad (4.A.5)$$

for $\omega \neq 0$, it holds that

$$E[|\phi(\omega)|^2] = O\left(\frac{1}{L^2}\right). \quad (4.A.6)$$

Furthermore,

$$E[\phi^*(\omega)\psi(\omega)] = \frac{1}{L^{3/2}} E \left[\left(\sum_{k=1, k: \omega_k \neq \omega}^K \alpha_k \mathbf{h}^H(\omega) \mathbf{a}_M(\omega_k) \cdot \sum_{l=0}^{L-1} e^{i(\omega_k - \omega)l} \right) \cdot \left(\mathbf{h}^H(\omega) \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} \mathbf{w}(l) e^{-i\omega l} \right) \right] = O\left(\frac{1}{L^{3/2}}\right) \quad (4.A.7)$$

where the last equality holds since the first factor inside the expectation converges to a constant and the second factor is bounded (in a mean-square sense). Finally,

$$\begin{aligned} L \cdot E[|\psi(\omega)|^2] &\rightarrow \frac{1}{L} \mathbf{h}_0^H(\omega) \sum_{l=0}^{L-1} \sum_{k=0}^{L-1} E[\mathbf{w}_l(\omega) \mathbf{w}_k^H(\omega)] e^{-i\omega(l-k)} \mathbf{h}_0(\omega) \\ &= \frac{1}{L} \mathbf{h}_0^H(\omega) \sum_{l=0}^{L-1} (L - |l|) E[\mathbf{w}_k(\omega) \mathbf{w}_{k-l}^H(\omega)] e^{-i\omega l} \mathbf{h}_0(\omega) \\ &\rightarrow \mathbf{h}_0^H(\omega) \sum_{l=0}^{L-1} E[\mathbf{w}_k(\omega) \mathbf{w}_{k-l}^H(\omega)] e^{-i\omega l} \mathbf{h}_0(\omega) \\ &\rightarrow \mathbf{h}_0^H(\omega) (\mathbf{a}_M(\omega) \mathbf{a}_M^H(\omega) P(\omega)) \mathbf{h}_0(\omega) \\ &= P(\omega) \end{aligned} \quad (4.A.8)$$

as $L \rightarrow \infty$ which together with (4.A.4), (4.A.6) and (4.A.7) shows (4.7.3).

4.B Proof of Theorem 4.2

Throughout this proof we shall omit the frequency variable ω (whenever no confusion can occur) to simplify the notation. Let

$$\bar{\boldsymbol{\delta}} = \frac{1}{L} \sum_{l=0}^{L-1} \tilde{\mathbf{w}}_l(\omega) e^{-i\omega l} \quad (4.B.1)$$

$$\tilde{\boldsymbol{\delta}} = \frac{1}{L} \sum_{l=0}^{L-1} \tilde{\mathbf{w}}_l(\omega) e^{-i\omega l} \quad (4.B.2)$$

Then we have that

$$\hat{\alpha}_{\text{FB-Capon}} - \alpha = \frac{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M}. \quad (4.B.3)$$

In what follows, we use the symbol \approx to denote an ‘‘asymptotic equality’’ that holds to within a second-order approximation. A straightforward manipulation of (4.B.3)

yields

$$\begin{aligned}
& \hat{\alpha}_{\text{Capon}} - \alpha \\
&= \frac{\mathbf{a}_M^H (\hat{\mathbf{R}}_{fb}^{-1} - \mathbf{R}^{-1}) \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M} + \frac{\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M} \\
&= \frac{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} (\mathbf{R} - \hat{\mathbf{R}}_{fb}) \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M} + \mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \left[\frac{1}{\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M} - \frac{1}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} + \frac{1}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} \right] \\
&\approx -\frac{\mathbf{a}_M^H \mathbf{R}^{-1} (\hat{\mathbf{R}}_{fb} - \mathbf{R}) \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} + \frac{\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} + \frac{\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \mathbf{a}_M^H (\mathbf{R}^{-1} - \hat{\mathbf{R}}_{fb}^{-1}) \mathbf{a}_M}{(\mathbf{a}_M^H \hat{\mathbf{R}}_{fb}^{-1} \mathbf{a}_M)(\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)} \\
&\approx -\frac{\mathbf{a}_M^H \mathbf{R}^{-1} (\hat{\mathbf{R}}_{fb} - \mathbf{R}) \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} + \frac{\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M} + \frac{(\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \mathbf{a}_M^H \mathbf{R}^{-1} (\hat{\mathbf{R}}_{fb} - \mathbf{R}) \mathbf{R}^{-1} \mathbf{a}_M}{(\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2}
\end{aligned} \tag{4.B.4}$$

which, in turn, implies that

$$\begin{aligned}
& E[\hat{\alpha}_{\text{Capon}} - \alpha] \\
&\approx E \left[\frac{-(\mathbf{a}_M^H \mathbf{R}^{-1} \hat{\mathbf{R}}_{fb} \mathbf{R}^{-1} \bar{\boldsymbol{\delta}})(\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) + (\mathbf{a}_M^H \mathbf{R}^{-1} \hat{\mathbf{R}}_{fb} \mathbf{R}^{-1} \mathbf{a}_M)(\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}})}{(\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2} \right].
\end{aligned}$$

Next we note that

$$\begin{aligned}
\hat{\mathbf{R}}_{fb} &= \frac{1}{2L} \sum_{l=0}^{L-1} \left[(\alpha \mathbf{a}_M e^{i\omega l} + \bar{\mathbf{w}}_l(\omega)) (\alpha \mathbf{a}_M e^{i\omega l} + \bar{\mathbf{w}}_l(\omega))^H \right. \\
&\quad \left. + (\tilde{\alpha} \mathbf{a}_M e^{i\omega l} + \tilde{\mathbf{w}}_l(\omega)) (\tilde{\alpha} \mathbf{a}_M e^{i\omega l} + \tilde{\mathbf{w}}_l(\omega))^H \right] \\
&= |\alpha|^2 \mathbf{a}_M \mathbf{a}_M^H + \frac{1}{2} (\alpha^* \bar{\boldsymbol{\delta}} + \tilde{\alpha}^* \tilde{\boldsymbol{\delta}}) \mathbf{a}_M^H + \frac{1}{2} \mathbf{a}_M (\alpha \bar{\boldsymbol{\delta}}^H + \tilde{\alpha} \tilde{\boldsymbol{\delta}}^H) \\
&\quad + \frac{1}{2L} \sum_{l=0}^{L-1} \left[\bar{\mathbf{w}}_l(\omega) \bar{\mathbf{w}}_l^H(\omega) + \tilde{\mathbf{w}}_l(\omega) \tilde{\mathbf{w}}_l^H(\omega) \right]
\end{aligned} \tag{4.B.5}$$

We also remind the reader of the assumption that $w(n)$ has nonzero third order moments. Since

$$\tilde{\mathbf{w}}_l(\omega) = \mathbf{J} \bar{\mathbf{w}}_{L-l-1}^*(\omega) \tag{4.B.6}$$

we have

$$\begin{aligned}
\tilde{\boldsymbol{\delta}} &= \frac{1}{L} \sum_{l=0}^{L-1} \tilde{\mathbf{w}}_l(\omega) e^{-i\omega l} = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{J} \bar{\mathbf{w}}_{L-l-1}^*(\omega) e^{-i\omega l} \\
&= e^{-i\omega(L-1)} \mathbf{J} \left(\frac{1}{L} \sum_{k=0}^{L-1} \bar{\mathbf{w}}_k(\omega) e^{-i\omega k} \right)^* = e^{-i\omega(L-1)} \mathbf{J} \bar{\boldsymbol{\delta}}^*
\end{aligned} \tag{4.B.7}$$

By using these facts, along with (4.B.5), we can write

$$\begin{aligned}
& (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2 \cdot E [\hat{\alpha}_{\text{Capon}} - \alpha] \\
& \approx E \left[(\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) \left(-\frac{1}{2} \mathbf{a}_M^H \mathbf{R}^{-1} (\alpha^* \bar{\boldsymbol{\delta}} + \tilde{\alpha}^* \tilde{\boldsymbol{\delta}}) (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \right. \right. \\
& \quad \left. \left. - \frac{1}{2} (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\alpha \bar{\boldsymbol{\delta}}^H + \tilde{\alpha} \tilde{\boldsymbol{\delta}}^H) \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \right) \right. \\
& \quad \left. + (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \left(\frac{1}{2} \mathbf{a}_M^H \mathbf{R}^{-1} (\alpha^* \bar{\boldsymbol{\delta}} + \tilde{\alpha}^* \tilde{\boldsymbol{\delta}}) (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) \right. \right. \\
& \quad \left. \left. + \frac{1}{2} (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\alpha \bar{\boldsymbol{\delta}}^H + \tilde{\alpha} \tilde{\boldsymbol{\delta}}^H) \mathbf{R}^{-1} \mathbf{a}_M \right) \right] \\
& = \frac{1}{2} E \left[\alpha^* (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}})^2 (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) + \tilde{\alpha}^* (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) (\mathbf{a}_M^H \mathbf{R}^{-1} \tilde{\boldsymbol{\delta}}) (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) \right. \\
& \quad \left. + \alpha (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) \left| \mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \right|^2 + \tilde{\alpha} (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) (\tilde{\boldsymbol{\delta}}^H \mathbf{R}^{-1} \mathbf{a}_M) \right. \\
& \quad \left. - \alpha^* (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}})^2 - \tilde{\alpha}^* (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\mathbf{a}_M^H \mathbf{R}^{-1} \tilde{\boldsymbol{\delta}}) (\mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \right. \\
& \quad \left. - \alpha (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2 (\bar{\boldsymbol{\delta}}^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) - \tilde{\alpha} (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2 (\tilde{\boldsymbol{\delta}}^H \mathbf{R}^{-1} \tilde{\boldsymbol{\delta}}) \right] \\
& = \frac{1}{2} \alpha (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) E \left[\left| \mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \right|^2 - (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\bar{\boldsymbol{\delta}}^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \right] \tag{4.B.8}
\end{aligned}$$

By the Cauchy-Schwartz inequality, the quantity between the curly parentheses in (4.B.8) is negative and so is its expectation. Hence part (a) of the theorem follows.

The bias for the forward-only Capon spectral estimate, $\hat{\alpha}_{\text{F-Capon}}$, can be obtained by replacing the forward-backward sample covariance \mathbf{R}_{fb} in (4.B.3)–(4.B.5) by the forward-only sample covariance matrix \mathbf{R}_f , and following the lines of (4.B.8). The result is as follows (to a second-order approximation):

$$\begin{aligned}
& (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M)^2 E [\hat{\alpha}_{\text{F-Capon}} - \alpha] \\
& \approx \alpha (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) E \left[\left| \mathbf{a}_M^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}} \right|^2 - (\mathbf{a}_M^H \mathbf{R}^{-1} \mathbf{a}_M) (\bar{\boldsymbol{\delta}}^H \mathbf{R}^{-1} \bar{\boldsymbol{\delta}}) \right]. \tag{4.B.9}
\end{aligned}$$

Hence to within a second-order approximation, the bias of the forward-backward Capon is indeed one half that of the forward-only Capon, which proves part (b) of the theorem.

Finally, to show part (c) of the theorem, note that

$$\begin{aligned}
\hat{\alpha}_{\text{APES}} - \alpha &= \frac{\mathbf{a}_M^H \hat{\mathbf{Q}}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \hat{\mathbf{Q}}^{-1} \mathbf{a}_M} \\
&\approx -\frac{\mathbf{a}_M^H \mathbf{Q}^{-1} (\hat{\mathbf{Q}} - \mathbf{Q}) \mathbf{Q}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{Q}^{-1} \mathbf{a}_M} + \frac{\mathbf{a}_M^H \mathbf{Q}^{-1} \bar{\boldsymbol{\delta}}}{\mathbf{a}_M^H \mathbf{Q}^{-1} \mathbf{a}_M} \\
&\quad + \frac{(\mathbf{a}_M^H \mathbf{Q}^{-1} \bar{\boldsymbol{\delta}}) \mathbf{a}_M^H \mathbf{Q}^{-1} (\hat{\mathbf{Q}} - \mathbf{Q}) \mathbf{Q}^{-1} \mathbf{a}_M}{(\mathbf{a}_M^H \mathbf{Q}^{-1} \mathbf{a}_M)^2} \tag{4.B.10}
\end{aligned}$$

Therefore,

$$\begin{aligned} (\mathbf{a}_M^H \mathbf{Q}^{-1} \mathbf{a}_M)^2 E[\hat{\alpha}_{\text{APES}} - \alpha] &\approx E \left[-(\mathbf{a}_M^H \mathbf{Q}^{-1} \hat{\mathbf{Q}} \mathbf{Q}^{-1} \bar{\boldsymbol{\delta}})(\mathbf{a}_M^H \mathbf{Q}^{-1} \mathbf{a}_M) \right. \\ &\quad \left. + (\mathbf{a}_M^H \mathbf{Q}^{-1} \hat{\mathbf{Q}} \mathbf{Q}^{-1} \mathbf{a}_M)(\mathbf{a}_M^H \mathbf{Q}^{-1} \bar{\boldsymbol{\delta}}) \right] \end{aligned} \quad (4.B.11)$$

Next note that

$$\begin{aligned} \hat{\mathbf{Q}} &= \hat{\mathbf{R}} - \mathbf{G}\mathbf{G}^H = \hat{\mathbf{R}} - \frac{1}{2}(\bar{\mathbf{g}}\bar{\mathbf{g}}^H + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^H) \\ &= |\alpha|^2 \mathbf{a}_M \mathbf{a}_M^H + \frac{1}{2} \alpha^* \bar{\boldsymbol{\delta}} \mathbf{a}_M^H + \frac{1}{2} \tilde{\alpha}^* \tilde{\boldsymbol{\delta}} \mathbf{a}_M^H + \frac{1}{2} \alpha \mathbf{a}_M \bar{\boldsymbol{\delta}}^H + \frac{1}{2} \tilde{\alpha} \mathbf{a}_M \tilde{\boldsymbol{\delta}}^H \\ &\quad + \frac{1}{2L} \sum_{l=0}^{L-1} (\bar{\mathbf{w}}_l(\omega) \bar{\mathbf{w}}_l^H(\omega) + \tilde{\mathbf{w}}_l(\omega) \tilde{\mathbf{w}}_l^H(\omega)) - |\alpha|^2 \mathbf{a}_M \mathbf{a}_M^H - \frac{1}{2} \alpha \mathbf{a}_M \bar{\boldsymbol{\delta}}^H - \frac{1}{2} \alpha^* \bar{\boldsymbol{\delta}} \mathbf{a}_M^H \\ &\quad - \frac{1}{2} \tilde{\alpha} \mathbf{a}_M \tilde{\boldsymbol{\delta}}^H - \frac{1}{2} \tilde{\alpha}^* \tilde{\boldsymbol{\delta}} \mathbf{a}_M^H - \frac{1}{2} (\bar{\boldsymbol{\delta}} \bar{\boldsymbol{\delta}}^H + \tilde{\boldsymbol{\delta}} \tilde{\boldsymbol{\delta}}^H) \\ &= \frac{1}{2L} \sum_{l=0}^{L-1} (\bar{\mathbf{w}}_\omega(l) \bar{\mathbf{w}}_\omega^H(l) + \tilde{\mathbf{w}}_\omega(l) \tilde{\mathbf{w}}_\omega^H(l)) - \frac{1}{2} (\bar{\boldsymbol{\delta}} \bar{\boldsymbol{\delta}}^H + \tilde{\boldsymbol{\delta}} \tilde{\boldsymbol{\delta}}^H) \end{aligned} \quad (4.B.12)$$

Using again the assumption of zero third-order moments of $w(n)$ and combining (4.B.11) and (4.B.12) yields

$$E\{\hat{\alpha}_{\text{APES}} - \alpha\} \approx 0 \quad (4.B.13)$$

and the proof is complete.

4.C Fast Matrix-Vector Multiplication

Let N_0 be the smallest power-of-two integer larger than or equal to N , and let \bar{N}_0 be defined similarly. Define the $N_0 \bar{N}_0 \times 1$ vector

$$\begin{aligned} \mathbf{x} &\triangleq \left[(\text{vec}(\mathbf{X}))^T \quad \mathbf{0}_{1, N_0 \bar{N}_0 - N \bar{N}} \right]^T \\ &= \left[X(0, 0) \quad \cdots \quad X(N-1, 0) \right. \\ &\quad \left. \cdots \quad X(0, \bar{N}-1) \quad \cdots \quad X(N-1, \bar{N}-1) \quad \mathbf{0}_{1, N_0 \bar{N}_0 - N \bar{N}} \right]^T \end{aligned} \quad (4.C.1)$$

where \mathbf{X} is the $N \times \bar{N}$ data matrix. Furthermore, let $\bar{\mathbf{Y}}$ be the (uniquely defined) $N_0 \bar{N}_0 \times N_0 \bar{N}_0$ left circulant matrix that has \mathbf{x} as its first column:

$$\bar{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{N_0 \bar{N}_0 - 2} & \mathbf{y}_{N_0 \bar{N}_0 - 1} \\ \mathbf{y}_1 & \mathbf{y}_2 & & \mathbf{y}_{N_0 \bar{N}_0 - 1} & \mathbf{y}_0 \\ \vdots & & \ddots & & \vdots \\ \mathbf{y}_{N_0 \bar{N}_0 - 2} & \mathbf{y}_{N_0 \bar{N}_0 - 1} & & \mathbf{y}_{N_0 \bar{N}_0 - 4} & \mathbf{y}_{N_0 \bar{N}_0 - 3} \\ \mathbf{y}_{N_0 \bar{N}_0 - 1} & \mathbf{y}_0 & \cdots & \mathbf{y}_{N_0 \bar{N}_0 - 3} & \mathbf{y}_{N_0 \bar{N}_0 - 2} \end{bmatrix}. \quad (4.C.2)$$

Next, define the $M\bar{M} \times \bar{M}N$ matrix

$$\mathbf{T}_1 \triangleq \begin{bmatrix} \mathbf{I}_M & \mathbf{0}_{M,L-1} & & & & \\ & & \mathbf{I}_M & \mathbf{0}_{M,L-1} & & \\ & & & & \ddots & \\ & & & & & \mathbf{I}_M & \mathbf{0}_{M,L-1} \end{bmatrix} \quad (4.C.3)$$

(all elements not shown are equal to zero) and the $\bar{L}N \times L\bar{L}$ matrix

$$\mathbf{T}_2 \triangleq \begin{bmatrix} \mathbf{I}_L & & & & \\ \mathbf{0}_{M-1,L} & & & & \\ & \mathbf{I}_L & & & \\ & \mathbf{0}_{M-1,L} & & & \\ & & \ddots & & \\ & & & \mathbf{I}_L & \\ & & & \mathbf{0}_{M-1,L} \end{bmatrix}. \quad (4.C.4)$$

In (4.C.4), $\mathbf{0}_{P,R}$ denotes a $P \times R$ matrix of zeros, and \mathbf{I}_P stands for the $P \times P$ identity matrix. Also, in the 1D case, \mathbf{T}_1 and \mathbf{T}_2 are identity matrices. Let

$$\bar{\mathbf{T}}_1 \triangleq [\mathbf{T}_1 \quad \mathbf{0}_{M\bar{M},N_0\bar{N}_0-\bar{M}N}] \quad (4.C.5)$$

and

$$\bar{\mathbf{T}}_2 \triangleq \begin{bmatrix} & \mathbf{T}_2 \\ \mathbf{0}_{N_0\bar{N}_0-\bar{L}N,L\bar{L}} & \end{bmatrix}. \quad (4.C.6)$$

Then

$$\mathbf{u}^H \mathbf{Y} = \mathbf{u}^H \bar{\mathbf{T}}_1 \bar{\mathbf{Y}} \bar{\mathbf{T}}_2 \quad (4.C.7)$$

for any vector \mathbf{u} , where \mathbf{Y} is defined in (4.5.4). To see this, consider the following $M \times L$ Hankel matrix:

$$\begin{aligned} \mathbf{Z}_{\bar{m},\bar{l}} &\triangleq \begin{bmatrix} X(0, \bar{m} + \bar{l}) & X(1, \bar{m} + \bar{l}) & \cdots & X(L-2, \bar{m} + \bar{l}) & X(L-1, \bar{m} + \bar{l}) \\ X(1, \bar{m} + \bar{l}) & X(2, \bar{m} + \bar{l}) & & X(L-1, \bar{m} + \bar{l}) & X(L, \bar{m} + \bar{l}) \\ \vdots & & \ddots & & \vdots \\ X(M-2, \bar{m} + \bar{l}) & X(M-1, \bar{m} + \bar{l}) & & X(N-3, \bar{m} + \bar{l}) & X(N-2, \bar{m} + \bar{l}) \\ X(M-1, \bar{m} + \bar{l}) & X(M, \bar{m} + \bar{l}) & \cdots & X(N-2, \bar{m} + \bar{l}) & X(N-1, \bar{m} + \bar{l}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Y}_{(\bar{m}+\bar{l})N} & \mathbf{Y}_{(\bar{m}+\bar{l})N+1} & \cdots & \mathbf{Y}_{(\bar{m}+\bar{l})N+L-2} & \mathbf{Y}_{(\bar{m}+\bar{l})N+L-1} \\ \mathbf{Y}_{(\bar{m}+\bar{l})N+1} & \mathbf{Y}_{(\bar{m}+\bar{l})N+2} & & \mathbf{Y}_{(\bar{m}+\bar{l})N+L-1} & \mathbf{Y}_{(\bar{m}+\bar{l})N+L} \\ \vdots & & \ddots & & \vdots \\ \mathbf{Y}_{(\bar{m}+\bar{l})N+M-2} & \mathbf{Y}_{(\bar{m}+\bar{l})N+M-1} & & \mathbf{Y}_{(\bar{m}+\bar{l})N+N-3} & \mathbf{Y}_{(\bar{m}+\bar{l})N+N-2} \\ \mathbf{Y}_{(\bar{m}+\bar{l})N+M-1} & \mathbf{Y}_{(\bar{m}+\bar{l})N+M} & \cdots & \mathbf{Y}_{(\bar{m}+\bar{l})N+N-2} & \mathbf{Y}_{(\bar{m}+\bar{l})N+N-1} \end{bmatrix} \end{aligned} \quad (4.C.8)$$

and note that

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Z}_{0,0} & \cdots & \mathbf{Z}_{0,\bar{L}-1} \\ \vdots & \ddots & \vdots \\ \mathbf{Z}_{\bar{M}-1,0} & \cdots & \mathbf{Z}_{\bar{M}-1,\bar{L}-1} \end{bmatrix} \quad (4.C.9)$$

which leads to (4.C.7) after some calculation.

Next, let Υ be the $N_0\bar{N}_0 \times N_0\bar{N}_0$ unitary DFT matrix whose (k, l) th element is equal to:

$$\Upsilon_{k,l} = \exp\left(-i2\pi \frac{(k-1)(l-1)}{N_0\bar{N}_0}\right). \quad (4.C.10)$$

Since $\bar{\mathbf{Y}}$ is a left circulant matrix, it follows that (see, e.g., [60, Chap. 8])

$$\Delta \triangleq \Upsilon \bar{\mathbf{Y}} \Upsilon \quad (4.C.11)$$

is a diagonal matrix. Hence

$$\mathbf{u}^H \mathbf{Y} = \mathbf{u}^H \bar{\mathbf{T}}_1 \Upsilon \Delta \Upsilon \bar{\mathbf{T}}_2 \quad (4.C.12)$$

for any vector \mathbf{u} , where all matrix-vector products can be evaluated by using the fast Fourier transform (FFT). It follows that the product $\mathbf{u}^H \mathbf{Y}$ can be computed in $O(N\bar{N} \log(N\bar{N}))$ floating point operations. The products $\mathbf{Y}\mathbf{u}$, $\mathbf{u}^H \tilde{\mathbf{Y}}$ and $\tilde{\mathbf{Y}}\mathbf{u}$ can be treated similarly.

4.D Fast Multiplication of Trigonometric Polynomials

Let n_0 be the smallest power-of-two integer larger than $n_p + n_q$, and let \bar{n}_0 be defined similarly. Furthermore, let

$$\begin{aligned} P(\eta, \bar{\eta}) &= \sum_{k=0}^{n_p-1} \sum_{\bar{k}=0}^{\bar{n}_p-1} p_{n_p-1-k, \bar{n}_p-1-\bar{k}}^* \cdot \exp\left(-i2\pi \left(\frac{k\eta}{n_0} + \frac{\bar{k}\bar{\eta}}{\bar{n}_0}\right)\right) \\ Q(\eta, \bar{\eta}) &= \sum_{k=0}^{n_q-1} \sum_{\bar{k}=0}^{\bar{n}_q-1} q_{k, \bar{k}} \cdot \exp\left(-i2\pi \left(\frac{k\eta}{n_0} + \frac{\bar{k}\bar{\eta}}{\bar{n}_0}\right)\right) \\ R(\eta, \bar{\eta}) &= \sum_{k=0}^{n_p+n_q-2} \sum_{\bar{k}=0}^{\bar{n}_p+\bar{n}_q-2} r_{k-n_p+1, \bar{k}-\bar{n}_p+1} \cdot \exp\left(-i2\pi \left(\frac{k\eta}{n_0} + \frac{\bar{k}\bar{\eta}}{\bar{n}_0}\right)\right) \end{aligned} \quad (4.D.1)$$

be the 2D DFTs of the coefficient vectors in the lemma's statement. By the 2D inverse DFT theorem we have that

$$r_{k-n_p+1, \bar{k}-\bar{n}_p+1} = \frac{1}{n_0\bar{n}_0} \sum_{\eta=0}^{n_0-1} \sum_{\bar{\eta}=0}^{\bar{n}_0-1} R(\eta, \bar{\eta}) \cdot \exp\left(i2\pi \left(\frac{k\eta}{n_0} + \frac{\bar{k}\bar{\eta}}{\bar{n}_0}\right)\right). \quad (4.D.2)$$

Next note from (4.8.10), (4.8.11) and (4.8.12) that

$$r_{k, \bar{k}} = \sum_{l=0}^{n_p-1} \sum_{m=0}^{n_q-1} \left(\sum_{\bar{l}=0}^{\bar{n}_p-1} \sum_{\bar{m}=0}^{\bar{n}_q-1} p_{l, \bar{l}}^* q_{m, \bar{m}} \right) \quad (4.D.3)$$

which is recognized as a 2D convolution. Using (4.D.3) it can be verified that

$$R(\eta, \bar{\eta}) = P(\eta, \bar{\eta})Q(\eta, \bar{\eta}). \quad (4.D.4)$$

The proof is now concluded by observing that $r_{k, \bar{k}}$ can be computed in the following way:

1. Compute $P(\eta, \bar{\eta})$ and $Q(\eta, \bar{\eta})$ in (4.D.1) using the 2D FFT.
2. Compute $R(\eta, \bar{\eta})$ according to (4.D.4).
3. Compute $r_{k, \bar{k}}$ according to (4.D.2) by using the 2D (inverse) FFT.

The polynomials $\tilde{r}(\omega, \bar{\omega})$ and $\hat{r}(\omega, \bar{\omega})$ are treated in a similar way.

4.E Fast Cholesky Factorization of $\hat{\mathbf{R}}_{fb}^{-1}$

The fast Cholesky factorization of $\hat{\mathbf{R}}_{fb}^{-1}$ is based on the concept of *displacement rank* [61], [34]. Let Φ be an arbitrary positive definite $P \times P$ matrix, and let Δ_r be a $P \times P$ matrix with ones on its r th diagonal below the main diagonal, and zeros everywhere else. We say that the *displacement rank of index r* of the matrix Φ is equal to

$$\rho_r(\Phi) \triangleq \text{rank}(\Phi - \Delta_r \Phi \Delta_r^T). \quad (4.E.1)$$

If Φ is a $P \times P$ positive definite matrix with displacement rank of index r equal to ρ , then a simple eigen-decomposition shows that there exist a $P \times \rho$ matrix \mathbf{G} , which is necessarily of full column rank, and a unitary $\rho \times \rho$ diagonal matrix \mathbf{F} such that

$$\Phi - \Delta_r \Phi \Delta_r^T = \mathbf{G} \mathbf{F} \mathbf{G}^H. \quad (4.E.2)$$

The triplet $\{\mathbf{G}, \mathbf{F}, r\}$ will be referred to as a *displacement factorization* of \mathbf{R} . As an aside, note that the displacement factorization of a matrix Φ is not unique, but it determines the matrix uniquely. This follows from the Lyapunov theorem (see, e.g., [62]) since all eigenvalues of Δ_r are equal to zero.

A displacement factorization of the forward sample covariance matrix for the 1D case can be found in [63]. A corresponding result for the 2D case was first established in [32] and is stated next without proof.

Lemma 4.A: *Partition $\hat{\mathbf{R}}_{fb}$ as*

$$\hat{\mathbf{R}}_{fb} = \begin{bmatrix} \hat{\mathbf{R}}_{0,0} & \cdots & \hat{\mathbf{R}}_{0,M-1} \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{R}}_{M-1,0} & \cdots & \hat{\mathbf{R}}_{M-1,M-1} \end{bmatrix} \quad (4.E.3)$$

where each block $\hat{\mathbf{R}}_{\bar{m}_1, \bar{m}_2}$ is of dimension $M \times M$ (in the 1D case, $\hat{\mathbf{R}}_{\bar{m}_1, \bar{m}_2}$ are scalars), and let $\mathbf{Z}_{\bar{m}, \bar{l}}$ be defined by (4.C.8). Let

$$\mathbf{G} = \begin{bmatrix} \hat{\mathbf{R}}_{0,0}^{-1/2} & \mathbf{0}_{M,M} & \mathbf{0}_{M,L} & \mathbf{0}_{M,L} & \mathbf{0}_{M,L} & \mathbf{0}_{M,L} \\ \hat{\mathbf{R}}_{1,0} \hat{\mathbf{R}}_{0,0}^{-1/2} & \hat{\mathbf{R}}_{1,0} \hat{\mathbf{R}}_{0,0}^{-1/2} & \mathbf{JZ}_{\bar{M}-2,0} & \mathbf{JZ}_{\bar{M}-1, \bar{L}-1} & \mathbf{Z}_{0,0} & \mathbf{Z}_{1, \bar{L}-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{\mathbf{R}}_{\bar{M}-1,0} \hat{\mathbf{R}}_{0,0}^{-1/2} & \hat{\mathbf{R}}_{\bar{M}-1,0} \hat{\mathbf{R}}_{0,0}^{-1/2} & \mathbf{JZ}_{0,0} & \mathbf{JZ}_{1, \bar{L}-1} & \mathbf{Z}_{\bar{M}-2,0} & \mathbf{Z}_{\bar{M}-1, \bar{L}-1} \end{bmatrix} \quad (4.E.4)$$

where \mathbf{J}_M is a reversal matrix. Furthermore, let Ψ be the following $(2M + 4L) \times (2M + 4L)$ diagonal matrix:

$$\Psi = \begin{bmatrix} \mathbf{I}_M & & & & & \\ & -\mathbf{I}_M & & & & \\ & & \mathbf{I}_L & & & \\ & & & -\mathbf{I}_{2L} & & \\ & & & & \mathbf{I}_L & \end{bmatrix}. \quad (4.E.5)$$

Then $\{\mathbf{G}, \Psi, M\}$ is a displacement factorization of $\hat{\mathbf{R}}_{fb}$ (in particular this implies $\rho_M(\hat{\mathbf{R}}_{fb}) = 2M + 4L$ in the 2D case; hence $\rho_M(\hat{\mathbf{R}}_{fb}) = 6$ in the 1D case).

An interesting consequence of Lemma 4.A is that $\hat{\mathbf{R}}_{fb}$ is fully parameterized by $O(M\bar{M}N)$ parameters and can hence be stored using $O(M\bar{M}N)$ bits of memory ($O(\bar{M})$ in the 1D case). This is so since the displacement factorization determines the covariance matrix uniquely (cf. the remark following (4.E.2)).

The article [32] also showed the following useful extension of Lemma 4.A.

Lemma 4.B: Let \mathbf{G} and Ψ be defined as in Lemma 4.A, and let

$$\hat{\mathbf{G}} = \begin{bmatrix} \hat{\mathbf{R}}_{0,0}^{-1/2} & \hat{\mathbf{R}}_{0,0}^{-1/2} & \mathbf{0}_{M,4L} \\ \mathbf{0}_{(\bar{M}-1)M,M} & \mathbf{0}_{(\bar{M}-1)M,M} & \mathbf{0}_{(\bar{M}-1)M,4L} \end{bmatrix}. \quad (4.E.6)$$

Furthermore, let $\tilde{\mathbf{G}} = [\mathbf{G}^T \quad \hat{\mathbf{G}}^T]^T$. Then $\{\tilde{\mathbf{G}}, \Psi, M\}$ is a displacement factorization of the matrix

$$\tilde{\mathbf{R}} \triangleq \begin{bmatrix} \hat{\mathbf{R}}_{fb} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}. \quad (4.E.7)$$

By using the technique in Section 4.C, the blocks $\hat{\mathbf{R}}_{m,0}$ in (4.E.4), and hence the displacement factorizations in Lemmas 4.A and 4.B, can be computed from the data in $O(M^3\bar{M} + MN\bar{N} \log(N\bar{N}))$ operations. This is so since only the first M columns, or equivalently the first M rows, of $\hat{\mathbf{R}}_{fb}$ are needed.

Having obtained the factorizations of $\hat{\mathbf{R}}_{fb}$, [32] demonstrated that we can obtain the columns of the $\hat{\mathbf{R}}_{fb}^{-1/2}$ by applying the *generalized Schur algorithm* [33], [34], to the factorization of $\hat{\mathbf{R}}$ in Lemma 4.B. This result is stated next.

Theorem 4.A: Let Φ be an arbitrary positive definite $P \times P$ matrix. Assume that $\{\mathbf{G}, \Psi, r\}$ is a displacement factorization of the matrix

$$\begin{bmatrix} \Phi & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (4.E.8)$$

where \mathbf{G} is $2P \times \rho$. Then the following procedure yields a sequence of vectors $\{\hat{\mathbf{l}}_k\}_{k=1}^P$ which equal the columns of $\Phi^{-1/2}$.

Start with $\mathbf{G}_1 = \mathbf{G}$ and perform the following steps for $k = 1, \dots, P$:

1. Let \mathbf{g}_k^H be the top row of \mathbf{G}_k .

2. Put

$$\mathbf{l}_k = \frac{\mathbf{G}_k \Psi \mathbf{g}_k}{\mathbf{g}_k^H \Psi \mathbf{g}_k} \quad (4.E.9)$$

and let $\hat{\mathbf{l}}_k$ be the P last elements of \mathbf{l}_k .

3. Let \mathbf{G}_{k+1} be defined through

$$\begin{bmatrix} \times_{1,\rho} \\ \mathbf{G}_{k+1} \end{bmatrix} = \mathbf{G}_k - \mathbf{l}_k \mathbf{g}_k^H + \begin{bmatrix} \mathbf{0}_{\min(r, P-k+1), 1} & \cdots & \mathbf{0}_{\min(r, P-k+1), 1} \\ (\mathbf{l}_k \mathbf{g}_k^H)_{1,1} & \cdots & (\mathbf{l}_k \mathbf{g}_k^H)_{1,\rho} \\ \vdots & & \vdots \\ (\mathbf{l}_k \mathbf{g}_k^H)_{P-k-r+1,1} & \cdots & (\mathbf{l}_k \mathbf{g}_k^H)_{P-k-r+1,\rho} \\ \mathbf{0}_{r,1} & \cdots & \mathbf{0}_{r,1} \\ (\mathbf{l}_k \mathbf{g}_k^H)_{P-k+2,1} & \cdots & (\mathbf{l}_k \mathbf{g}_k^H)_{P-k+2,\rho} \\ \vdots & & \vdots \\ (\mathbf{l}_k \mathbf{g}_k^H)_{P-r,1} & \cdots & (\mathbf{l}_k \mathbf{g}_k^H)_{P-r,\rho} \end{bmatrix} \quad (4.E.10)$$

where $\times_{P,R}$ denotes a $P \times R$ block of no interest.

Note that the application of the algorithm to our problem, given the displacement factorization $\{\tilde{\mathbf{G}}, \Psi, M\}$ in Lemma 4.B, requires $O(M^2 \bar{M}^2 N)$ floating point operations.

Finally we note also that a *stabilized* version of the generalized Schur algorithm, operating at approximately the same complexity as the generalized Schur algorithm presented above, has been proposed in [64]; however, no numerical instability problems have been observed in the experiments performed to date.

REFERENCES

- [1] J. D. Scargle, *Astronomical Time Series*, ch. “Astronomical Time Series Analysis”. Kluwer Academic Publishers, 1997. Eds.: D. Maoz et al.
- [2] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Norwell, MA: Kluwer Academic Publishers, 1996.
- [3] S. R. DeGraaf, “SAR imaging via modern 2-D spectral estimation methods,” *IEEE Trans. Image Processing*, vol. 7, pp. 729–761, May 1998.
- [4] F. Gini and F. Lombardini, “Multilook APES for multibaseline SAR interferometry,” *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1800–1803, 2002.
- [5] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice Hall, 1997.
- [6] S. M. Kay, *Modern Spectral Estimation, Theory and Application*. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [7] S. L. Marple, Jr., *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [8] R. O. Schmidt, “A signal subspace approach to multiple emitter location and spectral estimation,” Ph.D. thesis, Stanford University, Stanford, CA, USA, 1981.
- [9] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, “Exact and Large Sample ML Techniques for Parameter Estimation and Detection in Array Processing.” In *Radar Array Processing*, S. Haykin *et. al.* (eds.), Chapter 4. New York: Springer-Verlag, 1993.
- [10] J. Capon, “Maximum-likelihood spectral estimation,” in *Nonlinear Methods of Spectral Analysis* (S. Haykin, ed.), New York:Springer-Verlag, 1983.
- [11] J. Li and P. Stoica, “An adaptive filtering approach to spectral estimation and SAR imaging,” *IEEE Trans. Signal Processing*, vol. 44, pp. 1469–1484, June 1996.

- [12] P. Stoica, H. Li, and J. Li, "A new derivation of the APES filter," *IEEE Signal Processing Letters*, vol. 6, pp. 205–206, Aug. 1999.
- [13] L. M. Novak, G. R. Benitz, G. J. Owirka, and L. A. Bessette, "ATR performance using enhanced resolution SAR," *SPIE Proc. Optical Engineering in Aerospace Sensing*, Orlando, FL, pp. 332–337, Apr. 1996.
- [14] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, fourth ed., 2002.
- [15] P. D. Welch, "The use of fast Fourier transform for estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio and Electroacoustics*, vol. 15, pp. 70–73, June 1967.
- [16] E. G. Larsson, P. Stoica, and J. Li, "Spectral estimation via adaptive filter-bank methods: A unified analysis and a new algorithm," *Signal Processing*, vol. 82, pp. 1991–2001, Dec. 2002.
- [17] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 1985.
- [18] A. Jakobsson, *Model-Based and Matched-Filterbank Signal Analysis*. PhD thesis, Uppsala University, Uppsala, Sweden, 2000.
- [19] A. Jakobsson, T. Ekman, and P. Stoica, "Computationally efficient 2-D spectral estimation," in *Proc. Intl. Conf. on Acoust., Speech, Signal Processing*, 2000.
- [20] T. Ekman, A. Jakobsson, and P. Stoica, "On efficient implementation of the CAPON algorithm," in *Proc. European Signal Processing Conference (EUSIPCO)*, (Tampere, Finland), 2000.
- [21] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Trans. Signal Processing*, vol. 46, pp. 1954–1966, July 1998.
- [22] P. Stoica, H. Li, and J. Li, "Amplitude estimation of sinusoidal signals: survey, new results and an application," *IEEE Trans. Signal Processing*, vol. 48, pp. 338–352, Feb. 2000.
- [23] P. Stoica, A. Jakobsson, and J. Li, "Matched-filterbank interpretation of some spectral estimators," *Signal Processing*, vol. 66, no. 1, pp. 45–59, 1998.
- [24] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," vol. 37, pp. 8–15, Jan. 1989.
- [25] Y.-H. Choi, "On conditions for the rank restoration in forward/backward spatial smoothing," *IEEE Trans. Signal Processing*, vol. 50, pp. 2900–2901, Nov. 2002.

- [26] M. Jansson and P. Stoica, "Forward-only and forward-backward sample covariances – a comparative study," *Signal Processing*, vol. 77, no. 3, pp. 235–245, 1999.
- [27] A. Jakobsson and P. Stoica, "Combining Capon and APES for estimation of spectral lines," *Circuits, Systems and Signal Processing*, vol. 19, no. 2, pp. 159–169, 2000.
- [28] A. Jakobsson, S. L. Marple, Jr., and P. Stoica, "Two-dimensional CAPON spectral analysis," *IEEE Trans. Signal Processing*, vol. 48, pp. 2651–2661, Sept. 2000.
- [29] B. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 4, pp. 1333–1335, 1985.
- [30] Z.-S. Liu, H. Li, and J. Li, "Efficient implementation of CAPON and APES for spectral estimation," *IEEE Trans. Aerospace and Electron. Syst.*, vol. 34, pp. 1314–1319, Oct. 1998.
- [31] M. R. Palsetia and J. Li, "Using APES for interferometric SAR imaging," *IEEE Trans. Image Processing*, vol. 7, pp. 1340–1353, Sept. 1998.
- [32] E. G. Larsson and P. Stoica, "Fast implementation of two-dimensional APES and CAPON spectral estimators," *Multidimensional Systems and Signal Processing*, vol. 13, pp. 35–54, Jan. 2002.
- [33] T. Boros, A. H. Sayed, H. Lev-Ari, and T. Kailath, "A generalized Schur-type algorithm for the joint factorization of a structured matrix and its inverse," *Calcolo*, vol. 33, pp. 131–145, 1996.
- [34] T. Kailath and A. Sayed, *Fast Reliable Algorithms for Matrices with Structure*. Philadelphia: SIAM, 1999.
- [35] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1989.
- [36] E. G. Larsson, P. Stoica, and J. Li, "Amplitude spectrum estimation for two-dimensional gapped data," *IEEE Trans. Signal Processing*, vol. 50, pp. 1343–1354, June 2002.
- [37] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: John Wiley & Sons, 1968.
- [38] J. A. C. Lee and D. C. Munson, Jr., "Spatially variant apodization for image reconstruction from partial Fourier data," *IEEE Trans. Image Processing*, vol. 9, pp. 1914–1925, Nov. 2000.
- [39] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proc. IEEE*, vol. 71, pp. 917–925, August 1983.

- [40] D. A. Ausherman, A. Kozma, J. L. Walker, H. M. Jones, and E. C. Poggio, "Developments in radar imaging," *IEEE Trans. Aerospace and Electron. Syst.*, vol. 20, pp. 363–400, July 1984.
- [41] Y. Hua, "High resolution imaging of continuously moving object using stepped frequency radar," *Signal Processing*, vol. 35, pp. 33–40, January 1994.
- [42] I. J. Gupta, "High-resolution radar imaging using 2-D linear prediction," *IEEE Trans. Antennas and Propagation*, vol. 42, pp. 31–37, January 1994.
- [43] S. R. DeGraaf, "Sidelobe reduction via adaptive FIR filtering in SAR imagery," *IEEE Trans. Image Processing*, vol. 3, pp. 292–301, May 1994.
- [44] J. Li, P. Stoica, and D. Zheng, "An efficient algorithm for two-dimensional frequency estimation," *Multidimensional Systems and Signal Processing*, vol. 7, pp. 151–178, April 1996.
- [45] D. J. Andersh, M. Hazlett, S. W. Lee, D. D. Reeves, D. P. Sullivan, and Y. Chu, "XPATCH: a high-frequency electromagnetic scattering prediction code and environment for complex three-dimensional objects," *IEEE Antennas and Propagation Magazine*, vol. 36, pp. 65–69, Feb. 1994.
- [46] H. M. Adorf, "Interpolation of irregularly sampled data series – a survey," in *Astronomical Data Analysis Software and Systems IV*, vol. 77, pp. 460–463, 1995.
- [47] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplements*, vol. 15, pp. 417–426, 1974.
- [48] U. J. Schwarz, "Mathematical-statistical description of the iterative beam removing technique (CLEAN)," *Astronomy and Astrophysics*, vol. 65, pp. 345–356, 1978.
- [49] R. H. Chan and M. K. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Review*, vol. 38, pp. 427–482, Sept. 1996.
- [50] T. P. Bronez, "Spectral estimation of irregularly sampled multidimensional processes by generalized prolate spheroidal sequences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1862–1873, Dec. 1988.
- [51] I. K. Fodor and P. B. Stark, "Multitaper spectrum estimation for time series with gaps," *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp. 3472–3483, 2000.
- [52] E. G. Larsson and J. Li, "SAR image construction from periodically gapped phase-history data," in *Proc. SPIE Aerosense Conference*, (Orlando, FL), pp. 154–165, Apr. 2002.

- [53] P. Stoica, E. G. Larsson, and J. Li, "Adaptive filterbank approach to restoration and spectral analysis of gapped data," *The Astronomical Journal*, vol. 120, pp. 2163–2173, Oct. 2000.
- [54] E. G. Larsson, G. Liu, P. Stoica, and J. Li, "High-resolution SAR imaging with angular diversity," *IEEE Trans. Aerospace and Electronic Systems*, vol. 37, pp. 1359–1372, Oct. 2001.
- [55] T. Söderström and P. Stoica, *System Identification*. Hemel Hempstead, UK: Prentice Hall International, 1989.
- [56] E. G. Larsson, *The ubiquitous signal processing: applications to communications, spectral analysis and array processing*. PhD thesis, Uppsala University, Uppsala, Sweden, 2001.
- [57] A. Jakobsson, T. Ekman, and P. Stoica, "CAPON and APES spectrum estimation for real-valued signals," in *Proc. IEEE Digital Signal Processing Workshop*, Aug. 1998.
- [58] P. Stoica and T. Sundin, "Nonparametric NMR spectroscopy," *Journal of Magnetic Resonance*, vol. 152, pp. 57–69, 2001.
- [59] Z.-S. Liu, R. Wu, and J. Li, "Complex ISAR imaging of maneuvering targets via the Capon estimator," *IEEE Trans. Signal Processing*, vol. 47, pp. 1262–1271, May 1999.
- [60] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River, NJ: Prentice-Hall, 2000.
- [61] T. Kailath and A. H. Sayed, "Displacement structure: Theory and applications," *SIAM Review*, vol. 37, pp. 297–386, 1995.
- [62] W. J. Rugh, *Linear System Theory*. Upper Saddle River, NJ: Prentice-Hall, 1996.
- [63] J. Chun, T. Kailath, and H. Lev-Ari, "Fast parallel algorithms for QR and triangular factorization," *SIAM Journal of Scientific and Statistical Computation*, vol. 8, no. 6, pp. 899–913, 1987.
- [64] S. Chandrasekaran and A. H. Sayed, "Stabilizing the generalized Schur algorithm," *SIAM J. Matrix Analysis and its Applications*, vol. 17, pp. 950–983, 1996.
- [65] R. Wu, Z.-S. Liu, and J. Li, "Time-varying complex spectral analysis via recursive APES," *IEE Proc. – Radar, Sonar and Navigation*, vol. 145, pp. 354–360, Dec. 1998.

