ACTA UNIVERSITATIS UPSALIENSIS

Uppsala Dissertations from the Faculty of Science and Technology 22
Distributor: Uppsala University Library, Box 510, SE-751 20, Sweden

Andreas Jakobsson

Model-Based and Matched-Filterbank Signal Analysis

Dissertation in Signal Processing to be publicly examined in room 1111, Center of Mathematics and Information Technology (MIC), on February 11, 2000, at 10.15 a.m. for the Degree of Doctor of Philosophy. The examination will be conducted in English.

ABSTRACT


The dissertation deals with model-based and matched-filterbank signal analysis. The matched-filterbank (MAFI) spectral estimation approach is introduced, and it is shown that both the amplitude spectrum Capon (ASC) and the amplitude and phase estimation (APES) spectral estimators can be expressed as MAFI spectral estimators. A combined estimation procedure for data with mixed spectrum is introduced, as well as ASC and APES implementations for real-valued data. Computationally efficient implementations of the 2-D power spectrum Capon (PSC) and the 1-D and 2-D ASC are proposed.

An asymptotic Cramér-Rao bound for line-spectra estimation is derived. It is shown that the non-linear least squares method (NLSM) will asymptotically achieve the same statistical performance as the maximum likelihood method (MLM) even in the colored noise case. Sufficient conditions for identifiability are derived for known and unknown waveforms received through a multipath channel. Statistically efficient subspace-based estimators for the estimation of the time-delay and Doppler parameters are presented.

Keywords: non-parametric spectral estimation, efficient implementations, array processing, parameter estimation, identifiability, subspace fitting.

Andreas Jakobsson, Department of Information Technology, Uppsala University, PO Box 27, SE-751 08 Uppsala, Sweden.
Dissertation for the Degree of Doctor of Philosophy in Signal Processing presented at Uppsala University in 2000

ABSTRACT


The dissertation deals with model-based and matched-filterbank signal analysis. The matched-filterbank (MAFI) spectral estimation approach is introduced, and it is shown that both the amplitude spectrum Capon (ASC) and the amplitude and phase estimation (APES) spectral estimators can be expressed as MAFI spectral estimators. A combined estimation procedure for data with mixed spectrum is introduced, as well as ASC and APES implementations for real-valued data. Computationally efficient implementations of the 2-D power spectrum Capon (PSC) and the 1-D and 2-D ASC are proposed.

An asymptotic Cramér-Rao bound for line-spectra estimation is derived. It is shown that the non-linear least squares method (NLSM) will asymptotically achieve the same statistical performance as the maximum likelihood method (MLM) even in the colored noise case. Sufficient conditions for identifiability are derived for known and unknown waveforms received through a multipath channel. Statistically efficient subspace-based estimators for the estimation of the time-delay and Doppler parameters are presented.

Keywords: non-parametric spectral estimation, efficient implementations, array processing, parameter estimation, identifiability, subspace fitting.

Andreas Jakobsson, Department of Information Technology, Uppsala University, PO Box 27, SE-751 03 Uppsala, Sweden.

© Andreas Jakobsson 1999

ISSN 1104–2516

Printed in Sweden by Elanders Gotab, Stockholm, 1999
Distributor: Uppsala University Library, Box 510, SE–751 20, Sweden
To Ylva
Acknowledgments

The unpaved road of research is an occasionally manic-depressive path, full of thorns and futile attempts, but also one of joyous moments of intoxicating bliss. At times, it seems to me that one needs a somewhat masochistic mind to even consider treading it. I am deeply indebted to my supervisor Prof. Petre Stoica for acting as my guide on this long walk, and for making it a more rewarding and pleasant one. Petre is a native in the deep jungle of past and present literature and the vast marshes of worthy tasks and frustrating dead-ends, always possessing an unheard-of reference and a worthwhile answer to every question. His excellent guidance and expert knowledge has saved me from entering on many a deceitful stray path.

Furthermore, Prof. Torsten Söderström has my gratitude for supporting me and accepting me as a graduate student. I am also grateful to my co-authors: Dr. David Astély, M.Sc. Torbjörn Ekman, Dr. Hongbin Li, Prof. Jian Li, Dr. Lawrence Marple Jr., Prof. Lee Swindlehurst and Dr. Claes Tidestav. Having been able to share your thoughts and insights has been both fruitful and rewarding. It has been a pleasure working with you. I thank Prof. Sabine van Huffel, Prof. Arogyaswami Paulraj, and Prof. Lee Swindlehurst for inviting me to visit their research groups, as well as Prof. Miguel Lagunas for acting as my faculty opponent.

I also wish to thank everyone, past and present, at the Systems and Control Group and the Signals and Systems Group for creating such a pleasant work environment and such a rewarding way of preventing me from actually working. Our numerous social endeavors have been a constant source of most welcome distraction. I especially owe M.Sc. Torbjörn Ekman, Dr. Claes Tidestav and Dr. Joakim Sorelius; good friends who have made a big difference these last couple of years. Torbjörn also has my gratitude for once, while being dubiously attached to a 10.5 mm rope 200 m up a vertical rock in Utah, in the middle of a vicious thunderstorm, being amazingly cheerful about it.

Finally, I would like to express my deepest gratitude and appreciation to my fiancée Ylva for her unconditional love and support, without which this dissertation would never have been begun, nor ever completed.
## Contents

1 Introduction .............................................. 1
   1.1 Background and Overview of the Thesis ............... 1
   1.2 Matched-Filterbank Signal Analysis .................. 2
      1.2.1 The Periodogram Spectral Estimator ............ 3
      1.2.2 The Capon Spectral Estimator .................. 6
      1.2.3 Matched-Filterbank Spectral Estimation ........ 9
      1.2.4 On Efficient Implementations .................. 13
      1.2.5 The CAPES Spectral Estimator ................. 13
      1.2.6 Real-Valued Data ................................ 15
      1.2.7 Two-Dimensional Spectral Estimation ........... 16
   1.3 Model-Based Signal Analysis .......................... 18
      1.3.1 The Sinusoidal Data Model ..................... 18
      1.3.2 Asymptotic Results for Cisoid Parameter Estimation 19
      1.3.3 Multipath Identifiability ...................... 20
      1.3.4 Estimation Using a More Refined Data Model ..... 21
   1.4 Thesis Outline and Contributions ..................... 24
      1.4.1 Matched-Filterbank Signal Analysis ............ 24
      1.4.2 Model-Based Signal Analysis .................... 27
   1.5 Topics for Future Research .......................... 29
   1.6 Acronyms ............................................. 31
I  Matched-Filterbank Signal Analysis  33

2  Matched-Filterbank Interpretation of the Capon and APES Spectral Estimators  35
   2.1  Introduction ............................................. 35
   2.2  The Filterbank Approach ................................. 36
   2.3  The Matched-Filterbank Approach ...................... 38
      2.3.1  Capon ................................................. 39
      2.3.2  APES ................................................ 40
      2.3.3  MAFI ................................................ 41
   2.4  Spectrum Estimation .................................... 42
   2.5  Comparisons and Discussions ........................... 43
      2.5.1  The Computational Issue ............................. 43
      2.5.2  The Statistical Performance Issue .................. 44
   2.6  Numerical Examples ..................................... 46
      2.6.1  Estimation Performance Versus SNR ................ 46
      2.6.2  Estimation Performance Versus Filterlength ....... 48
   2.7  Conclusions .............................................. 51
   2.A  Proof of (2.22) ...................................... 52
   2.B  The Matrix Inversion Lemma ............................ 52
   2.C  Proof of Theorem 2.1 .................................. 53
   2.D  Proof of (2.48), (2.49) ................................. 55

3  On Efficient Implementation of the Capon Algorithm  59
   3.1  Introduction ............................................. 59
   3.2  The Matched Filterbank Approach ...................... 60
   3.3  Proposed Efficient Implementation ..................... 62
   3.4  Numerical examples ..................................... 64
   3.5  Conclusions .............................................. 68

4  Combining Capon and APES for Estimation of Spectral Lines  69
   4.1  Introduction ............................................. 69
   4.2  The Matched Filterbank Approach ...................... 71
4.3 Weighted Least-Squares ........................................ 72
4.4 CAPES ......................................................... 73
4.5 Numerical Examples ............................................ 74
4.6 Conclusions .................................................... 80

5 Capon and APES Spectrum Estimation for Real-Valued Signals 81
5.1 Introduction ..................................................... 81
5.2 The Real-Valued MAFI Approach .............................. 83
5.3 Bias Analysis .................................................... 85
5.4 Numerical Examples ............................................ 87
5.5 Conclusions .................................................... 88

6 Two-Dimensional Capon Spectral Analysis 89
6.1 Introduction ..................................................... 89
6.2 Problem Formulation ........................................... 91
6.3 Proposed Algorithm ............................................ 93
6.4 The Yule-Walker Equations ................................... 96
   6.4.1 The Toeplitz-Block-Toeplitz Estimate .................... 97
   6.4.2 The FB Block-Toeplitz Estimate ......................... 97
   6.4.3 The Outer Product Estimate ............................... 98
6.5 The 2-D Lattice Algorithm ................................... 98
6.6 Numerical Examples .......................................... 101
6.7 Conclusions .................................................... 106
6.A The Gohberg-Heinig formula .................................. 107
6.B Proof of Theorem 6.1 .......................................... 108

7 On Efficient Implementation of the 2-D Capon Algorithm 113
7.1 Introduction ..................................................... 113
7.2 The Matched Filterbank Approach ............................ 115
7.3 Proposed Efficient Implementation ........................... 118
7.4 Numerical Examples .......................................... 119
II Model-Based Signal Analysis

8 Cisoid Parameter Estimation in the Colored Noise Case 127
  8.1 Introduction and Preliminaries .......................... 127
  8.2 MLM and CRB ............................................. 130
  8.3 NLSM ...................................................... 135
  8.4 Numerical Examples ....................................... 137
    8.4.1 Estimation Performance versus SNR ............. 138
    8.4.2 Estimation Performance versus Sample Length ... 141
    8.4.3 Estimation of the Asymptotic CRB ............... 144
    8.4.4 Estimation Performance for Real Radar Data ..... 145

9 On the Identifiability of Multipath Parameters 149
  9.1 Multipath Parameter Estimation Problem ............... 149
  9.2 Derivation of the PSD Model ............................. 150
  9.3 Identifiability Properties of the PSD Model .......... 152

10 Subspace-Based Estimation of Time Delays and Doppler Shfts 155
  10.1 Introduction ............................................. 155
  10.2 Modeling ................................................ 158
    10.2.1 Assumptions ....................................... 158
    10.2.2 Connections with DOA Estimation ................ 161
    10.2.3 Identifiability ..................................... 162
  10.3 ML Estimation and the CRB .............................. 163
  10.4 Subspace-Based Estimation Methods .................... 165
    10.4.1 Noise Subspace Fitting ............................ 166
    10.4.2 Signal Subspace Fitting ........................... 168
    10.4.3 MUSIC .............................................. 170
    10.4.4 ESPRIT ............................................. 171
  10.5 Numerical examples .................................... 174
10.5.1 Performance versus SNR ................. 174
10.5.2 Closely spaced sources ................. 176
10.5.3 Large Doppler shifts ..................... 178
10.5.4 Resolving closely-spaced arrivals ....... 178
10.6 Conclusions ................................ 178
"Where shall I begin, please your Majesty?" he asked.
"Begin at the beginning," the King said, very gravely,
"and go on till you come to the end: then stop."
\textit{Lewis Carroll}

Chapter 1

Introduction

This thesis is concerned with model-based and matched-filterbank (MAFI) signal analysis. In the first part of the thesis, we examine non-parametric signal analysis, in particular spectrum estimation, using matched-filterbank based approaches, whereas in the second part we study model-based, or parametric, signal analysis making use of a sinusoidal signal model. The present chapter is meant to serve as an introduction to the main topics considered in the thesis, as well as to give an overview of the material presented.

1.1 Background and Overview of the Thesis

Signal analysis, and in particular the subtopic of spectrum analysis, is an important data analysis tool which has found applications in a wide variety of fields, including speech processing, telecommunications, radar and sonar systems, biomedical and seismic signal processing, and economics.

There are two broad classes of approaches to signal analysis: \textit{the non-parametric approaches} and \textit{the parametric approaches}. While the non-parametric approaches typically do not postulate any model for the observed data, the parametric approaches do assume some (normally \textit{a priori} known) model so that the signal is represented by a set of parameters. As a result, for parametric methods, the signal analysis problem is reduced to that of estimating the parameters of the given model. Parametric methods may offer a more accurate signal description than non-parametric methods, provided that the data indeed does observe the model assumed. In the more likely case where an accurate data model is not available, the parametric methods will be sensitive to model mismatch. In such cases the more robust non-parametric methods, which make no assumption on the
data, may be a better choice. This fact has of lately created a renewed interest in the topic of non-parametric signal analysis.

This thesis is concerned with both non-parametric and parametric methods to signal analysis. In the first part of this thesis we study non-parametric signal analysis, in particular non-parametric spectrum estimation using matched-filterbank (MAFI) based approaches. The problem of interest is to describe how the power of some observed data is distributed along the frequency axis, the so-called power spectral density (PSD). In the second part, we will instead focus on parametric signal analysis using a sinusoidal signal model. We study estimation approaches, estimation limits and the topic of identifiability, i.e., if it is possible to estimate the parameter set of a given model.

Which of the two approaches that should be used in any specific case depends primarily on the application of interest, and how much that is known about it. In some applications the choice is obvious; in others far from easy. As we are here primarily interested in the different approaches, not in the different applications as such, we present no guidelines as to which approach should be used for any specific application.

1.2 Matched-Filterbank Signal Analysis

The concept of non-parametric signal analysis is to extract some kind of information concerning some observed data, without assuming any knowledge of the underlying data model that has generated it. In general, this formulation is too vague and too general to be of any real use. For instance, one may ask what kind of information is really relevant and of interest for any given problem? The answer lies fully in the problem at hand, and there are no general truths. For some problems one might be interested in the mean and the variance of the observed data, in others the maximum or the minimum values. In this thesis, we will focus on the problem of extracting information about the PSD of the data, i.e., the problem of describing how the power of the observed data is distributed along the frequency axis.

This is one of the most classical estimation problems in signal processing, and as such there is a vast amount of literature published on the topic. Any attempt to fully cover the field is thus doomed to fail, and for this reason the scope of any study must, even in this subfield, by necessity be rather focused. So is also this work, where we will mainly limit our attention on the so-called filterbank approach to spectral estimation.

The filterbank approach is a very general approach that estimates the desired spectrum using a bank of bandpass filters, each filter centered
around a given frequency of interest. This is depicted in Figure 1.1. By calculating the power in a given filter's output, one obtains an estimate of the PSD at the center frequency of that filter.

![Diagram](image)

**Figure 1.1:** The filterbank approach to PSD estimation

Before we immerse ourselves in the details on how one should proceed to design the individual bandpass filters in the filterbank in an appropriate way, we feel the need to remind the reader of the origin of the very word spectrum. The word dates back to Sir Isaac Newton who in 1671 introduced it to describe the band of light he observed when passing sunlight through a glass prism [New71]. The word is a variant of the Latin word “specter”, meaning image or ghostly apparition, and the reader should thus not be overly concerned if some of the expressions in the following might seem somewhat scary.

### 1.2.1 The Periodogram Spectral Estimator

The first, most well-known and most widely used spectral estimator is the Periodogram, originally derived by Sir Arthur Schuster in 1898 to determine “hidden periodicities” (non-obvious periodic signals) in time series [Sch98]. Schuster applied the Periodogram to find hidden periodicities in the monthly sunspot numbers for the years 1749 to 1894 [Sch06]. These are shown in Figure 1.2. His analysis yielded an estimate of 11.125 years for the sunspot cycle, and this is the basis for the classic 11-year sunspot cycle referenced in astronomical literature. This 11-year cycle is marked as a dashed line in Figure 1.3, where some different spectral estimates for the sunspot data are shown. As can be seen in the figure, there are clearly other “hidden periodicities” in the data - although it is hard to actually state how many periodicities there really are. The problem of estimating
how many periodicities that are actually “hidden” in a data set is in itself a vast and highly interesting field of research, alas beyond the scope of this thesis. We refer the interested reader to [Fuc88] and the references therein for further information on this topic.

Figure 1.3 also illustrates the problem one is faced with when examining real data. Without knowing the underlying data model, it is futile to exhaust oneself in trying to evaluate how well the different methods depict the true spectrum. This evaluation can really only be done by actually possessing knowledge concerning how the data was constructed. For this reason, we will in the remainder of this thesis focus our attention primarily on the study of artificially constructed data sets.

Before proceeding into the details of the different spectral estimators, it should also be noted that the estimation of an observed signal’s spectrum based on a finite number of observations is normally an ill-posed problem from a statistical point of view, as it requires an estimation of an infinite number of independent spectral values based on a finite number of observations. An often reasonable assumption made by most non-parametric estimation methods is to assume that the PSD of the observed data is approximately constant over a narrow band around any given frequency. This assumption is also made in this text.

The Periodogram forms an estimate of the PSD, \( \hat{\phi}(\omega) \), as

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

where \( x(t) \) denotes the observed data, \( N \) the number of collected samples, and \( \omega \) the frequency of interest. Although the Periodogram, as well as its various variations including the famous Blackman-Tuckey’s method [BT59], were not designed as filterbank spectral estimators, they can under fairly weak assumptions be cast into the rather general framework of filterbank
Figure 1.3: Different spectral estimates for the sunspot data. (a) The Periodogram. (b) The windowed Periodogram. (c) The Capon spectrum. (d) The APES spectrum.

approaches. This can be seen by rewriting (1.1) as

$$
\hat{\phi}(\omega) = \frac{1}{\beta} \left| \sum_{k=0}^{N-1} h_k(\omega)x(N-k) \right|^2, \quad \beta = 1/N,
$$

where $\beta = 1/N$, and where the $k$th tap of the bandpass filter centered around the frequency $\omega$, $h_k(\omega)$, is defined as

$$
h_k(\omega) \triangleq \frac{1}{N} e^{i\omega k}, \quad k = 0, \ldots, N-1.
$$

By padding the filter sequence in (1.3) with zeros, the truncated convolution in (1.2) can be rewritten as

$$
x_F(N) = \sum_{k=0}^{\infty} h_k(\omega)x(N-k).
$$
As a result, the spectral estimate in (1.2) can be seen as being formed from a single sample of the filtered signal. From this viewpoint, it is not surprising that the variance of the Periodogram is rather large (see [SM97] for a further discussion on the Periodogram and on its filterbank interpretation).

1.2.2 The Capon Spectral Estimator

As the Periodogram was not designed as a filterbank method, one may ponder whether one could not find a better choice of the bandpass filters used. This is indeed so, and there exist several other methods which can significantly improve the performance of the estimates (see [SM97] and the references therein). One widely used approach is the REFIL (REfined FILter) method, also called the Thomson method after its inventor, which was first introduced in [Tho82] and then further developed in [MS91] (see also the discussion in [Bro92, OS93, RS95]). A detailed development of the method can be found in [SM97]. The REFIL method is conceptually close to that of the Daniell approach [Dan46] in its way to reduce the variance of the spectral estimates as compared to the Periodogram. In fact, both estimators can, under a local smoothness assumption, be interpreted as approximations of the maximum likelihood PSD estimator [SS99].

The idea behind REFIL is to design a bank of filters which pass the signal component within the passbands as much as possible relative to the total power, as well as attenuate the frequencies outside the corresponding passbands. The resulting estimator in general gives better spectral estimates than the traditional Fourier-based methods. Nevertheless, the REFIL filter, as well as the previously mentioned approaches, are all data-independent in the sense that the filters do not adapt in any way to the actually measured data. An alternative approach would be to make the filterbank's bandpass filters data-dependent, such that the filters are designed from the actual measured data. This intriguing idea was originally proposed by Capon [Cap69] for multi-dimensional seismic array frequency-wavenumber analysis. It was later reformulated by Lacoss [Lac71] for one-dimensional (1-D) time-series problems. Furthermore, Capon and Goodman [CG70, Cap71] showed that the Capon spectrum has a mean and a variance that behave like the averaged Periodogram, i.e., the Capon spectral estimate usually have a lower variance than the Periodogram.

The $L$-tap long Capon filter, $h_{\omega}$,

$$ h_{\omega} = [ \ h_0(\omega) \ \ldots \ h_{L-1}(\omega) \ ]^T, $$

(1.5)

where $(\cdot)^T$ denotes the transpose operator, is designed so that the bandpass filters try to minimize the total filter output power while passing the center
frequency $\omega$ undistorted, i.e.,

$$h_\omega = \arg \min_{h_\omega} \|R^{-1}h_\omega \| \quad \text{subject to} \quad h_\omega^*a_\omega = 1,$$

(1.6)

where $(\cdot)^*$ and $\hat{R}$ denote the complex conjugate transpose and the outer-product sample covariance matrix

$$\hat{R} \triangleq \frac{1}{M} \sum_{t=1}^{M} y(t)y^*(t),$$

(1.7)

with $M = N - L + 1$. Here, $y(t)$ denotes the $L$ most recent samples of $x(t)$, i.e.,

$$y(t) \triangleq \begin{bmatrix} x(t) & \ldots & x(t + L - 1) \end{bmatrix}^T, \quad (t = 1, \ldots, M)$$

(1.8)

and the Fourier vector, $a(\omega)$, is defined as

$$a_\omega \triangleq \begin{bmatrix} 1 & e^{i\omega} & \ldots & e^{i(L-1)\omega} \end{bmatrix}^T.$$  

(1.9)

Thus, the constraint $h_\omega^*a_\omega = 1$ in (1.6),

$$h_\omega^*a_\omega = \sum_{k=0}^{L-1} h_k^*(\omega)e^{ik\omega} = 1,$$

(1.10)

is nothing but a requirement that the filter $h_\omega$ has unit gain for the frequency of interest. The bandpass filter minimizing (1.6) is found as

$$h_\omega = \frac{\hat{R}^{-1}a_\omega}{a_\omega^*\hat{R}^{-1}a_\omega}.$$  

(1.11)

This filter occurs frequently in the literature, in many different applications, and has several different names. In the original paper by Capon it is named the Maximum Likelihood Method (MLM) [Cap69] and in more recent literature, such as [Mar87, Kay88], it is named the Minimum Variance Spectral Estimator (MVSE) or the Minimum Variance Method (MVM) [DeG98]. None of the names are really appropriate as the filter neither is a maximum likelihood (ML) spectral estimator nor does it possess the minimum variance property. In adaptive beamforming the same filter is named Linearly Constrained Minimum Variance (LCMV) [Fro72], or alternatively Minimum Variance Distortionless Response (MVDR) [Hay91]. In more recent research, the filter is more often named just the Capon filter [Li99], and this is the name we will use in the following.

The Capon spectral estimator, as obtained from the Capon filter, can be formulated in two different ways, by either estimating the power of the
filter output (as usually is done in filterbank approaches), here denoted the Power Spectrum Capon (PSC), i.e.,

\[
\hat{\varphi}_\omega^{PSC} = \frac{1}{M} \sum_{i=1}^{M} |h_\omega^* y(t)|^2 = h_\omega^* \hat{R} h_\omega = \frac{1}{a_\omega^* \hat{R}^{-1} a_\omega},
\]

(1.12)

or by estimating the amplitude of a complex-valued sinusoid (cisoid) located at the frequency of interest (as is done in Chapter 2), here denoted the Amplitude Spectrum Capon (ASC), i.e.,

\[
\hat{\varphi}_\omega^{ASC} = |h_\omega^* Y_\omega|^2 = \left| \frac{a_\omega^* \hat{R}^{-1} Y_\omega}{a_\omega^* \hat{R}^{-1} a_\omega} \right|^2
\]

(1.13)

where

\[
Y_\omega = \frac{1}{M} \sum_{i=1}^{M} y(t)e^{-j\omega t}.
\]

(1.14)

Note that neither of these estimators, which in general are different from each other, are really estimators of the power spectral density (PSD) of the data. As is shown in the next section, both estimators are actually estimating the amplitude spectrum of \(x(t)\) in a passband centered on \(\omega\). To obtain an estimate of the PSD from an estimate of the amplitude spectrum, one needs to divide the estimate by \(\beta\), where \(\beta\) denotes the frequency bandwidth of the filter \(h_\omega\) (see also Figure 1.1). This division by \(\beta\) is often omitted in the literature, but it is required to yield an estimate of the PSD.

As the filter \(h_\omega\) is data dependent, its bandwidth \(\beta\) is not necessarily data independent, nor is it necessarily frequency independent. Thus, the division by \(\beta\) may not only represent a simple scaling, but it may also change the shape of the resulting spectral estimate. There are various possibilities for determining the bandwidth \(\beta\), depending on the degree of precision we are aiming for. The simplest possibility is to set

\[
\beta = \frac{1}{L}.
\]

(1.15)

This choice is motivated by the time-bandwidth product, which states that the bandwidth of an \(L\)-tap filter should be roughly \(1/L\). An often more exact expression for \(\beta\) can be obtained as [LSGM86]

\[
\beta = \frac{a_\omega^* \hat{R}^{-2} a_\omega}{\left[ a_\omega^* \hat{R}^{-1} a_\omega \right]^2}.
\]

(1.16)

The motivation for this choice of \(\beta\) is that the (equivalent) bandwidth of a bandpass filter can be defined as the support of the rectangle centered
on $\omega$ (the filter’s center frequency) that concentrates the whole energy in the filter’s frequency response (see also the discussion in [SM97]). In the following, we will mainly study the amplitude spectral estimates; the problem of how the PSD is most appropriately estimated from an amplitude spectrum is in itself a challenging problem, alas beyond the scope of this thesis.

In the remainder of this thesis, we will occasionally denote the two estimators simply the Capon spectral estimator where it is clear from the context which of the versions we are referring to.

### 1.2.3 Matched-Filterbank Spectral Estimation

In Chapter 2, we make use of the fact that the filterbank approach basically reduces the problem of estimating the spectrum to that of estimating the amplitude of a sinusoidal signal buried in *colored noise* (see, e.g., [LS96a, LSGM86]). The filter output can thus be written as

$$h_{\omega}^* y(t) = \alpha_{\omega} e^{i \omega t} + \epsilon(t), \quad t = 1, \ldots, M, \quad \omega \in (0, 2\pi],$$

(1.17)

where $\alpha_{\omega}$ denotes the (complex-valued) amplitude of the sinusoidal signal referred to above and $\epsilon(t)$ is the noise term. We will in the following, somewhat erroneously, assume that $\epsilon(t)$ is *approximately* uncorrelated with $\alpha_{\omega} e^{i \omega t}$. This assumption is basically only true for spectral lines, although it has been found to be a reasonable approximation in other cases as well [LS96a]. Assuming that we know the bandpass filter, the least-squares estimate of the complex amplitude, $\alpha_{\omega}$, in (1.17) is found as

$$\hat{\alpha}_{\omega} = \frac{1}{M} \sum_{t=1}^{M} h_{\omega}^* y(t) e^{-i \omega t} = h_{\omega}^* Y_{\omega}. \quad (1.18)$$

The ASC estimator in (1.13) is thus nothing but an estimate of the squared complex amplitude, $|\hat{\alpha}_{\omega}|^2$. Similarly, by ignoring the additive noise term, $\epsilon(t)$, in (1.17), the PSC estimator in (1.12) can be seen as an estimate of $|\hat{\alpha}_{\omega}|^2$ as well. From this observation, it should come as no surprise that the ASC spectral estimator will yield estimates with better resolution, as well as often somewhat more accurate spectral amplitude estimates, than the PSC spectral estimator. This is especially true for cases with a low signal-to-noise ratio (SNR). An example of the methods’ estimates are illustrated in Figure 1.4.
The remaining problem thus lies in how the filters should be designed in an appropriate way. The filters can be designed in numerous ways, and as there is not any straightforward answer to what is really appropriate, any choice of design will in some sense be ad hoc. We proceed by choosing to design the filterbank bandpass filters as matched filters. By definition a matched filter should be designed so that the corresponding signal-to-noise ratio (SNR) in the filter’s output is maximized (see Chapter 2 for more details)

$$\max_{h_\omega} \frac{\text{Signal power}}{\text{Noise power}} = \max_{h_\omega} \frac{|h_\omega^* a_\omega|^2}{h_\omega^* \hat{Q} h_\omega}, \quad (1.19)$$

where $\hat{Q}$ denotes the estimated covariance matrix of the corrupting noise sequence. As the ratio in (1.19) is unchanged by a multiplication with any complex number, the optimization problem in (1.19) is equivalent to (cf. (1.6))

$$\min_{h_\omega} h_\omega^* \hat{Q} h_\omega \quad \text{subject to} \quad h_\omega^* a_\omega = 1. \quad (1.20)$$

The solution to the design problem in (1.19) is thus (cf. (1.11))

$$h_\omega = \frac{\hat{Q}^{-1} a_\omega}{a_\omega^* \hat{Q}^{-1} a_\omega}. \quad (1.21)$$

In Chapter 2, it is shown that both the previously discussed ASC spectral estimator and the recently proposed Amplitude and Phase Estimation

---

**Figure 1.4:** Estimated spectrum using (a) the PSC and (b) the ASC spectral estimator. The true spectrum is also shown (dotted).
(APES) estimator, which was originally derived by means of a relatively involved approximate maximum likelihood approach [LS96a], can be interpreted as members of this class of matched-filterbank (MAFI) methods. With the MAFI interpretation, the APES spectral estimator is given a significantly simpler and more intuitive derivation, which has since been extended to a form using forward-backward averaging [LSLJ97, LLS98, Li99], as well as to the case of real-valued data (see Chapter 5). The APES spectral estimator can also be derived from pure narrowband-filter design considerations [SLL99]. What differentiates the ASC and the APES spec-

\[ \hat{Q}_{ASC} = \hat{\mathbf{R}} \]
\[ \hat{Q}_{APES} = \hat{\mathbf{R}} - \mathbf{Y}_\omega \mathbf{Y}_\omega^* \]  

One of the important results shown in Chapter 2 is that the ASC estimator will underestimate the true spectrum, whereas the APES estimator is unbiased (to within a second-order approximation). An example of this can be seen in Figure 1.5 which shows an estimated line spectrum, using the two methods, compared to the true spectrum. The reason for the ASC estimator's poor performance in the case depicted in Figure 1.5 is, partly, explained in the following. Consider a case where the data consists of a number of sinusoidal components in noise with small power. The Capon

\[ \text{Figure 1.5: Estimated spectra using (a) the ASC and (b) APES spectrum estimate. The true spectrum is also shown (dotted).} \]
filter design will then tend to place nulls to annihilate the strong sinusoidal components, but will pay little attention to the weak noise component. The consequence is that the gain of the filter will be nearly zero at the sinusoid frequencies, but may take rather larger values at other frequencies (see, for example, Chapter 2 and the numerical examples in [LS96a], which demonstrate this behavior of the Capon filter). The APES filter design will, however, take also these weak noise components into consideration. An illustration of this is shown in Figure 1.6 for the example above.

![Graph](image)

**Figure 1.6:** The data dependent filter gain for (a) Capon and (b) APES, calculated at the second sinusoidal frequency. The true spectrum is also shown (dotted).

However, as the reader well knows a free lunch is indeed very rare, and the observant reader will notice that the APES spectral estimate in Figure 1.5(b) has somewhat wider peaks than the corresponding ASC estimate (see also Figure 1.7). Thus, the APES spectral estimate, *for a given filter-length* $L$, does sacrifice some resolution to gain better amplitude and phase estimates as compared with the ASC estimator. This might seem as a significant drawback of the APES estimator, but as is shown in Chapter 2 it is often possible to choose a larger filterlength, and thus gain in resolution, than it is with the ASC estimator. This fact does in some sense reduce the impact of the sacrificed resolution, and this drawback is deemed to be minor.
1.2.4 On Efficient Implementations

In many practical applications, the data source is often rapidly varying and the number of available *stationary* data samples is highly limited. The rapid data rate normally implies that the needed computational requirements to process a batch of data is a critical resource. Thus, one might simply not have the time to perform computationally highly complex operations (such as, for instance, a Singular Value Decomposition (SVD) or a Cholesky factorization). In such cases, the need of a computationally efficient algorithm is vital, and it is thus of significant interest to find more efficient implementations for any particular algorithm.

For this reason, we propose an efficient implementation of the ASC spectral estimator in Chapter 3. The implementation makes use of the highly structured problem formulation, and is based on efficient estimates of the linear prediction coefficients (LPC).

![Figure 1.7: A close-up of the spectral estimates in Figure 1.5. (a) The ASC spectrum. (b) The APES spectrum.](image)

1.2.5 The CAPES Spectral Estimator

In general, the APES estimator gives a significantly more accurate amplitude and phase estimate of the spectrum than the PSC and the ASC spectrum estimators do [LS96a, SJL98, Li99]. As was mentioned above, it was shown in Chapter 2 that ASC, to within a second-order approximation, will be biased (downward) whereas APES will be unbiased. For data con-
taining spectral peaks this means that at the true peak locations the APES estimator will outperform the PSC and the ASC spectrum estimators. As the peak locations are normally assumed to be unknown, the spectrum is estimated over a grid of frequencies and sinusoidal components are found as the spectral peaks.

However, as a result of its lower resolution, the peak locations in the APES spectral estimate are in general not located exactly at the true frequencies. This can also be seen by studying a close-up of Figure 1.5, as is shown in Figure 1.7. Note the very accurate APES estimates at the true frequencies. In Chapter 4, we show that the APES estimates of the peak locations are in many cases somewhat biased, and that the corresponding peak amplitude estimate in these cases will also be (slightly) biased. For a signal with a smooth spectrum, this is but a minor drawback, whereas for data containing strong sinusoidal components it might be of importance. It is further shown in Chapter 4 that the APES estimates will have clearly lower resolution than the ASC estimates. This can also be seen in Figure 1.7 by examining the peak resolution for the two peaks.

![Figure 1.8: The true spectrum for the data in Figure 1.9.](image)

To circumvent these disadvantages, we propose an alternative estimation procedure for data with a mixed spectrum containing an unknown number of spectral lines. The suggested method first estimates the sinusoidal peak locations using the ASC frequency estimator, i.e., the peak locations of the ASC spectrum, and thereafter refines the amplitude and phase estimates at these frequencies using APES. The resulting estimator, termed CAPES, is in Chapter 4 found to give significantly more accurate amplitude and phase estimates than both the ASC and the APES estimators. The improved resolution is also illustrated in Figure 1.9 for the spectrum shown in Figure 1.8 (see Section 4.5 for further details on this example).
Figure 1.9: Illustration of the resolution and accuracy for the different spectral estimators. Note the different scale for APES. (a) The Periodogram. (b) The APES spectrum. (c) The ASC spectrum. (d) The CAPES spectrum of the first 4 peaks.

### 1.2.6 Real-Valued Data

In real life, many signals are actually *real-valued* which the previous discussion of the Capon and the APES filter design does not exploit. For such signals, it ought to be more appropriate to design the filter taking into account the fact that the spectrum is *symmetric*. Indeed, it is reasonable
to expect a somewhat better performance of a real-valued filter design because by construction the filters will then pass both frequencies of interest, \( \omega \) and \(-\omega\), undistorted, whereas a complex-valued design will only pass \( \omega \) undistorted, and will try to null \(-\omega\), therefore yielding less power in the filtered output. In Chapter 5, we present real-valued versions of the ASC and the APES spectral estimators, and show that these will indeed have better performance than the complex-valued estimators (for real-valued data). Furthermore, we discuss the bias of these real-valued ASC and APES spectral estimators as compared to that of the complex-valued estimators. It is found that the complex-valued estimators will have a (slight) bias (at most frequencies) for real-valued data, whereas the real-valued estimators will not.

![Image](a)

![Image](b)

**Figure 1.10:** An example of an image obtained by (a) the 2-D Periodogram and (b) the 2-D PSC algorithm presented in Chapter 6 (see Section 6.6 for further details on this example).

### 1.2.7 Two-Dimensional Spectral Estimation

As was mentioned in Section 1.2.4, it is in many applications critical that one uses computationally efficient algorithms. This is especially so for multi-dimensional estimation problems, as then the computational burdens are often significantly higher than in the one-dimensional (1-D) counterparts. This observation also holds true for two-dimensional (2-D) spectral estimation.
The problem of 2-D high resolution spectral estimation has been widely studied in the past literature (see, e.g., [DM84, McC82, Mar87] and the many references therein), as well as in more recent contributions such as [LLS98, LLL98, Mar]. Applications occur in a wide variety of fields, such as image processing [Jai89], synthetic aperture radar (SAR) image formation and target feature extraction [DeG98, JWE+96, Ben98], motion compensation using interferometric SAR (ISAR) [LWL99, CH99, QC89], and 2-D nuclear magnetic resonance (NMR) [LRL98, HKB+89], to mention a few. In many of these applications, it is of key importance to obtain high resolution estimates, as for example it is in SAR image formation and target feature extraction, which are becoming increasingly important in many civilian and military applications [Kle97]. Popular approaches include the 2-D Periodogram and, in the higher resolution cases, the 2-D AR and the 2-D Capon spectral estimators. A number of approaches has been suggested for efficient estimation of the 2-D AR spectrum (see, e.g., [ML89, TES89, Mar87, PFGL98]), whereas only limited efforts have been made to simplify the 2-D Capon estimator [Mar87, LLL98].

In Chapter 6, we present a computationally efficient algorithm to compute the 2-D PSC, which proves to be significantly faster than previous attempts to speed up this computation [LLL98]. This work is extended in Chapter 7, where we present a computationally efficient algorithm to compute the 2-D ASC. Both these implementations, as well as the majority of the other methods in the literature, are based on the computation of the 2-D Fast Fourier Transform (FFT). How this computation should be done efficiently is in itself an important and challenging problem which is under constant scrutiny (see, e.g., [GM86, Sev99] and the references therein).
1.3 Model-Based Signal Analysis

In parametric signal analysis, some information of the underlying data model generating the observed signal is assumed to be known. Such a priori knowledge of the data model may be assumed for different reasons, but it is in no way an uncommon situation. Consider, for instance, a radar scenario where a known signal is transmitted at a given time. The measured backscattered signal can then be well modeled as a sum of delayed and attenuated copies of the transmitted signal, corrupted by some kind of additive noise (see, e.g., [Lev88]). If this knowledge of the data model is not taken into account, the problem of estimating the distance and velocity of the reflecting target will be unsolvable.

Although the PSD is often of key interest, the way the signal’s power is distributed over frequency is but one part of all the information that might be found. In parametric signal analysis, the problem of analyzing the signal is not really that of estimating the PSD, but instead the set of parameters that describes the data, where the PSD might or might not be included in this set. The parametric signal analysis problem is thus reduced into that of estimating the set of parameters that describe the measured data in the best possible way.

1.3.1 The Sinusoidal Data Model

One of the most common data models occurring in the literature is that the measured data, $x(t)$, is considered to be modeled as $d$ superimposed complex-valued sinusoidal components corrupted by additive noise, i.e.,

$$x(t) = \sum_{k=1}^{d} \alpha_k e^{i(\omega_k t + \varphi_k)} + e(t), \quad (t = 1, \ldots, N)$$ (1.24)

where $\alpha_k$, $\omega_k$ and $\varphi_k$ are the $k$th complex sinusoid’s (cisoloid’s) amplitude, frequency and initial phase, respectively, and $e(t)$ is an additive noise process. We will here assume that $\omega_k \in [-\pi, \pi]$ and that $\alpha_k > 0$ to avoid the otherwise possible phase ambiguity. It is common practice to assume that the number of sinusoids, $d$, are known. Often this is not so, and the problem of estimation this number is in itself an important research topic (see, e.g., [Fuc88]).

The model in (1.24) is often convenient to rewrite in vector form for the $L$ most recent samples. This is done by introducing $s_k(t) = \alpha_k e^{i(\omega_k t + \varphi_k)}$, and

$$a_\omega = \begin{bmatrix} 1 & e^{i\omega} & \ldots & e^{i(L-1)\omega} \end{bmatrix}^T$$ (1.25)
\[ A = \begin{bmatrix} a_{\omega_1} & \cdots & a_{\omega_d} \end{bmatrix}. \] (1.26)

Then
\[
\mathbf{y}(t) \triangleq \begin{bmatrix} x(t) \\ x(t + 1) \\ \vdots \\ x(t + L - 1) \end{bmatrix} = A_{\omega} \mathbf{s}(t) + \mathbf{e}(t) \quad (t = 1, \ldots, M) \] (1.27)

where \( M = N - L + 1 \), and

\[
\mathbf{s}(t) = \begin{bmatrix} s_1(t) & \cdots & s_d(t) \end{bmatrix}^T
\] (1.28)

\[
\mathbf{e}(t) = \begin{bmatrix} e(t) & \cdots & e(t + L - 1) \end{bmatrix}^T
\] (1.29)

\[
\omega = \begin{bmatrix} \omega_1 & \cdots & \omega_d \end{bmatrix}.
\] (1.30)

This data model may seem very specialized, but it is in fact very general and can be used to describe a wide variety of both temporal and spatial signal estimation problems [Mar87, Böh91, Kay93, SM97].

### 1.3.2 Asymptotic Results for Cisoid Parameter Estimation

As was mentioned previously, it is possible to obtain more accurate estimates by exploiting knowledge of the data model. The most intuitively appealing approach to do so, would be to choose our estimates as the values that are most likely to “describe” the observed data. Rephrased, choose the estimates that maximize the likelihood function given the observed data

\[
\hat{\Theta} = \arg \max_\Theta p(y; \Theta),
\] (1.31)

where \( \Theta \) denotes the parameter vector to be estimated. This is the famous maximum likelihood (ML) estimator, and it is well-known that it is asymptotically efficient under rather general conditions, i.e., it will in large samples attain the ultimate performance corresponding to the Cramér-Rao lower bound (CRB). The CRB is a theoretical lower bound for any unbiased estimator. It is given as the inverse Fisher information matrix (FIM),

\[
\text{CRB} = [\text{FIM}]^{-1},
\] (1.32)

where

\[
\text{FIM}_{ij} = -E \left( \frac{\partial^2 \ln p(y; \Theta)}{\partial \Theta_i \partial \Theta_j} \right),
\] (1.33)

with \( E(\cdot) \) denoting the expectation. For an in-depth discussion of the ML estimator and the CRB see, e.g., [Kay93, SM97].
Unfortunately, the ML estimates may in practice often be very hard to obtain. This is due to the severe computational complexity associated with the maximization over the entire parameter space in (1.31). Furthermore, the likelihood function has a complicated multimodal shape with a very sharp global maximum corresponding to the desired estimate [SMFS89], and as a result, it requires very accurate initialization.

Therefore, it is of particular interest to find computationally more attractive estimators that have a similar performance as the ML method. The non-linear least squares method (NLSM) is one such method. The NLSM (temporal or spatial) frequency estimates, $\hat{\omega}$, are obtained as (cf. (1.27))

$$
\hat{\omega} = \arg\min_{\omega} \| y(t) - A_\omega s(t) \|^2
$$

$$
= \arg\max_{\omega} \text{tr} \left\{ \Pi_{A_\omega} \hat{R} \right\},
$$

where $\Pi_{A_\omega} = A_\omega (A_\omega^* A_\omega)^{-1} A_\omega^*$ is the projection matrix onto the range space of $A_\omega$, and $\hat{R}$ is the sample covariance matrix given in (1.7). When the additive noise is white and Gaussian distributed, it is well-known that the NLSM in (1.34) can be interpreted as the ML estimate (see, e.g., [Kay93, SM97]). Furthermore, in Chapter 8, where the performance of the NLSM is examined, it is shown that the NLSM asymptotically achieves the same statistical performance as the ML method even in the colored noise case.

It is worth noting that, for any given asymptotical result, one might always ask how large $N/$SNR/etc should be to hold in a given “non-asymptotic” scenario? The answer can vaguely be said to be that the value of $N/$SNR/etc should be such that the estimation errors (for example) should be “reasonably small”. In the cases when this can not be assumed to hold, the estimation problem is most likely in the threshold region, and one is in trouble anyway. For an elaboration on this topic, we refer the reader to the interesting discussion found in [Tom98].

It should be mentioned here that, even though the NLSM is computationally significantly simpler than the ML method, it still requires accurate initialization. This is a common problem with the parametric estimators; the estimates are often difficult to obtain due to local minima.

### 1.3.3 Multipath Identifiability

A fundamental issue in parametric signal analysis often needed to be resolved before proceeding to the actual estimation of the parameter set modeling the data, is the issue of identifiability. If an estimation problem is not identifiable, it means that given the observed data it is theoretically not
possible to obtain a consistent\footnote{An estimate is said to be consistent if it will converge to the true value as the number of data samples grows without bound.} estimate of the parameter set modeling the data, independently of approach taken. Similarly, if an estimation problem is identifiable, it is not impossible to obtain a consistent estimate of the parameter set, i.e., it is not necessarily straightforward to obtain an accurate estimate of the parameter set, but it is at least not impossible.

![Sensor Reflecting target](image.png)

\textbf{Figure 1.11:} The low-angle radar data model.

In Chapter 9, we examine the issue of identifiability for a classical estimation problem in array processing, that of estimating the parameters of a received signal consisting of noise and an unknown waveform arriving at a sensor through both a direct path and an attenuated and delayed specular path. Applications making use of this particular data model can, for instance, be found in low-angle radar [BS]. An example of the low-angle radar data model is shown in Figure 1.11.

We show that a commonly-used frequency-domain model for such a signal is valid essentially only under the assumption that the waveform that underwent multipath propagation is band-limited. Furthermore, we derive sufficient conditions for the aforementioned model to be identifiable from second-order statistics (i.e., its parameters can be uniquely determined from the PSD of the received signal). We also show that in the frequently considered case of a white waveform, neither the attenuation coefficient, nor the waveform power or the noise power can be unambiguously determined.

### 1.3.4 Estimation Using a More Refined Data Model

In this section, we examine a case where the simple data model presented in Section 1.3.1 does not describe the measured data sufficiently well, and must as a result be extended to enable us to achieve the best performance possible.

In Chapter 10, the problem of estimating the delays and Doppler shifts from multiple reflections of moving targets is studied. Here, the transmitted signal waveform, \( s(t) \), and the number of multipath reflections, \( d \), are assumed to be known. This situation occurs, for instance, in radar appli-
Figure 1.12: Multiple reflections from moving targets.

In this case, the simple data model presented in Section 1.3.1 is not sufficient to describe the measured data, and must as a result be extended to incorporate this further knowledge. The received data vector, $\mathbf{x}(t)$, may be better modeled as

$$\mathbf{x}(t) = \sum_{k=1}^{d} \mathbf{a}_k s(t - \tau_k) e^{j\omega D_k t} + \mathbf{n}(t),$$

where the parameters $\tau_k$, $\omega D_k$, and $\mathbf{a}_k$ are the time-delay, frequency offset and spatial signature of the $k$th arrival. Following the idea first presented in [Swi98], the data model (1.35) is in Chapter 10 rewritten in matrix form as (cf. (1.27))

$$\mathbf{X}_t = \mathbf{Q}_t(\tau, \omega) \mathbf{A} + \mathbf{N}_t,$$

where

$$\tau = \begin{bmatrix} \tau_1 & \cdots & \tau_d \end{bmatrix}^T.$$
\[ \omega = \begin{bmatrix} \omega_{D_1} & \cdots & \omega_{D_d} \end{bmatrix}^T, \]

(1.38)

and \( Q_1(\tau, \omega) \) is a matrix formed from the known signal sequence as is described in Chapter 10. In the previously discussed data model, the matrix \( A_\omega \) is a known function of the parameters of interest, and

\[ S = \begin{bmatrix} s(1) & \cdots & s(M) \end{bmatrix} \]

is treated as an unknown unstructured matrix. On the other hand, in (1.36) it is \( Q_1 \) that is parameterized and \( A \) that is unstructured. In essence, the roles of time and space have thus been reversed.

The classical approach to estimate \( \tau_k \) and \( \omega_{D_k} \) is to form a matched filter that correlates the received signal, \( x(t) \), with a delayed and frequency shifted version of the known signal [Tre68, Hel95]. However, this estimate is efficient only if a single signal is received. In Chapter 10, we make use of the modified data model in (1.36) to develop efficient subspace-based estimators for the case when multiple arrivals are present. We also study under which conditions the estimation problem will be identifiable.
1.4 Thesis Outline and Contributions

This section is intended to give an overview of the thesis, and to summarize the contributions. The thesis consists of two parts. The first part is devoted to non-parametric signal analysis, primarily focused on matched-filterbank spectral estimation. The second part deals with parametric signal analysis, covering general data models.

All chapters are written so that they should be as self-containing as possible; this is done to help a reader interested in only a part of the material. As a result of this approach, some of the background material occurs more than once.

1.4.1 Matched-Filterbank Signal Analysis

The first part consists of six chapters and is concerned with non-parametric matched-filterbank (MAFI) spectral estimation. As the MAFI interpretation of Capon and APES is introduced in Chapter 2, it is preferable to read this chapter before proceeding to Chapter 3 - 5.

Chapter 2

In Chapter 2, the MAFI spectral estimation approach is treated, and it is shown that both the ASC and the APES spectral estimators can be expressed as MAFI spectral estimators. The MAFI interpretation of APES in this chapter leads to a significantly simpler and a more intuitive derivation than the one presented in [LS96a]. Furthermore, the MAFI interpretation provides further insights into the properties of and the relationship between the ASC and the APES spectral estimation methods, and it is shown that ASC, to within a second-order approximation, will be biased (downward) whereas APES will be unbiased.

The results of Chapter 2 have been published in part as


A related contribution, although not included in this thesis, is

Chapter 3

In Chapter 3, an efficient implementation of the ASC estimator is derived. The implementation, which is based on the Burg algorithm and the FFT, will be computationally significantly faster than the implementation recently proposed in [LLL98]. The implementation can be seen as a combination of the algorithm in [LLL98] and the efficient PSC algorithm proposed in [Mus85].

The results in Chapter 3 have been submitted for possible published as


Chapter 4

In Chapter 4, we show that the APES estimates of the peak locations are in many cases somewhat biased, and that the corresponding peak amplitude estimate in these cases will also be (slightly) biased. It is further shown that the APES estimates will have lower resolution than the ASC estimates. To circumvent these disadvantages, we propose an alternative estimation procedure for data with a mixed spectrum containing an unknown number of spectral lines. The suggested method first estimates the sinusoidal peak locations using the Capon frequency estimator, and thereafter refines the amplitude and phase estimates at these frequencies using APES. The resulting estimator, termed CAPES, is in Chapter 4 found to give significantly more accurate amplitude and phase estimates than both the ASC and the APES estimators.

The results of Chapter 4 have been published in part as


and are accepted for publication as

Chapter 5

In Chapter 5, we present real-valued versions of the ASC and the APES spectral estimators, and show that these will have better performance than the complex-valued estimators for real-valued data. Furthermore, we discuss the bias of these real-valued ASC and APES spectral estimators as compared to that of the complex-valued estimators. It is found that the complex-valued estimators will have a (slight) bias (at most frequencies) for real-valued data.

The results of Chapter 5 have been published in part as


Chapter 6

In Chapter 6, we present a computationally efficient algorithm for computing the 2-D PSC spectral estimator as well as a novel method to compute the forward prediction matrices. Furthermore, we discuss the difference in the so-obtained 2-D spectral estimate as compared to other published implementations, as well as the spectral estimate obtained by instead using the Whittle-Wiggins-Robinson algorithm to estimate the prediction matrices.

The results of Chapter 6 have been published in part as


and are submitted for possible publication as

Chapter 7

In Chapter 7, an efficient implementation of the 2-D ASC estimator is derived. The implementation, which will depend only on the (forward) linear prediction matrices and the (forward) prediction error covariance matrices, can be implemented using the 2-D FFT. The implementation is a 2-D extension of the algorithm proposed in Chapter 3. The results in Chapter 7 have been submitted for possible publication as


1.4.2 Model-Based Signal Analysis

The second part consists of three chapters and is concerned with parametric signal analysis. The chapters can be read in a stand-alone manner.

Chapter 8

In Chapter 8, we present an asymptotic CRB for line-spectra estimation, and examine the asymptotic performance of the ML and NLS estimators. It is well-known that when the additive noise is white and Gaussian distributed, the NLSM can be interpreted as the ML estimate. In Chapter 8 it is shown that the NLSM will asymptotically achieve the same statistical performance as the ML method even in the colored noise case. The results of Chapter 8 have been published in part as


Chapter 9

In Chapter 9, we examine the issue of identifiability for the problem of estimating the parameters of a received signal consisting of noise and an unknown waveform arriving at a sensor through both a direct path and
an attenuated and delayed specular path. We show that a commonly-used frequency-domain model for such a signal is valid essentially only under the assumption that the waveform that underwent multipath propagation is band-limited. Furthermore, we derive sufficient conditions for the aforementioned model to be identifiable from second-order statistics, as well as show that in the frequently considered case of a white waveform, neither the attenuation coefficient, nor the waveform power or the noise power can be unambiguously determined.

The results of Chapter 8 have been published in part as


**Chapter 10**

In Chapter 10, we examine the parameter estimation and identifiability of an extended data model. The data is assumed to consist of multiple time-delayed and Doppler shifted backscatters of a known signal. We present two statistically efficient subspace-based estimators for the estimation of the time-delay and Doppler parameters. Furthermore, we examine under which conditions the estimation problem will be identifiable, as well as present the ML estimator and the CRB.

The results of Chapter 10 have been published in part as


A related contribution, although not included in this thesis, is

1.5 Topics for Future Research

There are several interesting possibilities for further research that can be attempted with the material in this thesis as a starting point. In this section, we shall outline a number of such possible extensions.

- In Chapter 2, two different approaches for estimating the noise covariance matrix, \( \hat{Q} \), are suggested in the MAFI framework. These will as shown lead to the ASC and the APES spectral estimators. An open problem worthy of more careful examination is other possibilities to estimate \( \hat{Q} \) in a more "optimal" way (see also the discussion in Section 2.7).

Furthermore, we have observed that one might obtain superior spectral estimates, by combining the ASC and the APES noise covariance estimates in (1.22) and (1.23), i.e.,

\[
\hat{Q}_{\text{Combined}} = \hat{R} - \alpha \mathbf{Y}_\omega \mathbf{Y}_\omega^* \tag{1.39}
\]

for some constant \( \alpha \). The so-obtained spectral estimate will keep the better resolution of ASC and still gain some of the benefits of APES. An open question is how the constant \( \alpha \) should be chosen in an "optimal" manner, as well as a deeper insight into if and why such a combined estimator would be preferable.

- As is discussed in Section 1.2.2, a number of different approaches exist as to how the filter bandwidth, \( \beta \), should be computed in an appropriate way (see also [LSGM86, SM97]). This is a quite interesting topic in need of a thorough investigation.

- In [S JL96], we presented an extension of the classical matrix optimization result shown in (1.6) to the case of a singular or nearly singular covariance matrix. Such a situation would occur, for instance, for the filterbank spectral estimators when the filterlength \( L \geq N/2 \). Intuitively, the longer filterlengths should yield higher resolution spectral estimates. However, our experience of the use of the extended matrix optimization result for the filterbank spectral estimators is that this is not so. The spectral estimates so-obtained instead tend to be of a rather poor quality. Both the issue of why this is so, and what possible remedies that can be suggested are in need of careful examination.
• A computationally efficient real-valued ASC estimator is straightforwardly derived from the results in Chapter 3 and Chapter 5. An interesting application for such an algorithm can be found in, for instance, speech processing [KP95, MR].

• Another topic for future research would be to perform a closer examination as to why one obtains better spectral estimates by using the Burg algorithm (or the lattice algorithm presented in Chapter 6 in the 2-D case) as compared to the Levinson-Durbin algorithm (see, e.g., [UB75] for a related study on the 1-D AR spectral estimator). A related topic would be to examine why the ASC estimator in finite samples gives a better estimate than the PSC estimator.

• In [LWL99], a time recursive 2-D ASC spectral estimator was introduced for time-varying Interferometric SAR (ISAR) data. A straightforward improvement on the results presented there would be to use either of the closed form estimators presented in Chapter 6 or Chapter 7, together with a time-recursive version of the 2-D lattice algorithm introduced in Chapter 6.

Also note that, by applying the techniques used to derive the 2-D PSC estimator in Chapter 6, a more efficient 2-D ASC estimator than the one presented in Chapter 7 could be derived. The development ought to follow closely to the proof of Theorem 6.1.

• Another problem worth studying is the extension of the identifiability results presented in Chapter 9 to auto-regressive moving average (ARMA) sequences.

• Finally, it would be of interest to extend the results in Chapter 10 to the case of an observed unknown signal along the lines in [Ast99, SG99]. This extension should be rather straightforward, but can be made more challenging by extending the studied model so that a fixed frequency offset is added (cf. (1.35))

\[
x(t) = \sum_{k=1}^{d} a_k(t) s(t - \tau_k) e^{j(\omega_{D_k} + \omega_{D_0})t} + n(t),
\]

(1.40)

where \(\omega_{D_0}\) denotes the fixed frequency offset. In telecommunication, such a frequency offset might be due to a possible carrier offset. The model can be further generalized by allowing \(a_k(t)\) to be slowly time-varying.
1.6 Acronyms

1-D  One-Dimensional
2-D  Two-Dimensional
ACRB  Asymptotic Cramér-Rao Bound
APES  Amplitude and Phase Estimation
AR   Auto-Regressive
ASC  Amplitude Spectrum Capon
CAPES Capon frequency estimation combined with APES amplitude estimation
CRB  Cramér-Rao Bound
DOA  Direction of Arrival
FFT  Fast Fourier Transform
FIM  Fisher Information Matrix
FIR  Finite Impulse Response
ISAR  Interferometric Synthetic Aperture Radar
LCMV  Linearly Constrained Minimum Variance
LPC  Linear Prediction Coefficients
LS   Least Squares
MAFI Matched Filterbank
ML   Maximum Likelihood
MLM  Maximum Likelihood Method
MSE  Mean Squared Error
MVDR Minimum Variance Distortionless Response
MVSE Minimum Variance Spectral Estimator
NLS  Nonlinear Least-Squares
NLSM Nonlinear Least-Squares Method
NMR  Nuclear Magnetic Resonance
NSF  Noise Subspace Fitting
PSC  Power Spectrum Capon
PSD  Power Spectral Density
REFIL Refined Filter Method
RMSE Root Mean Square Error
SAR  Synthetic Aperture Radar
SNR  Signal-to-Noise Ratio
SSF  Signal Subspace Fitting
SVD  Singular Value Decomposition
ULA  Uniform Linear Array
WSF  Weighted Subspace Fitting
WWRA Whittle-Wiggins-Robinson Algorithm
Part I

Matched-Filterbank Signal Analysis
Chapter 2

Matched-Filterbank
Interpretation of the Capon
and APES Spectral
Estimators

We make use of a matched-filterbank (MAFI) approach to derive spectral estimators for stationary signals with mixed spectra. We show that the Capon spectral estimator as well as the more recently introduced APES (Amplitude and Phase EStimation) method are members of the MAFI class. A seemingly new MAFI-based spectral estimator is shown to reduce to APES as well. The MAFI interpretation of Capon and APES, along with some additional analysis, provides further insights into the properties of and the relationship between these two spectral estimation methods.

2.1 Introduction

The derivation of the Capon spectral estimator (see, e.g., [Cap69, Cap83, Lac71, LSGM86]) is quite related to that of a matched-filterbank processor. However, the fact that Capon is indeed a matched-filterbank spectral estimator does not appear to be widely known: to the best of our knowledge only the recent reference [Wax95] addressed this connection. In the present chapter we first introduce a matched-filterbank (MAFI) approach to spectral estimation for stationary signals with mixed (i.e., both continuous and discrete) spectra. Then we show that Capon is a member of the MAFI class of spectral estimators. We also show that the more recently introduced APES (Amplitude and Phase EStimation) method [LS96a] belongs
to the MAFI class as well. The belonging of APES to the MAFI class was conjectured in [LS96a].

The MAFI approach to spectral estimation may be used to devise new spectral estimation methods. Interestingly enough, a most reasonable implementation of a seemingly novel MAFI-based spectral estimator, derived in this chapter, is also shown to reduce to the APES method.

The MAFI interpretation of Capon and APES, along with some additional analysis, provides further insights into the properties of and the relation between these two spectral estimators. In particular, we prove that the APES spectral estimate is lower bounded by the Capon estimate at all frequencies. Additionally, we show by means of a higher-order expansion technique that the Capon estimator underestimates the true spectrum whereas the APES spectral estimator is unbiased (to within a second-order approximation). These theoretical results, supplemented with practical evidence that the Capon estimator is indeed biased downwards in samples of practical length whereas APES is nearly unbiased (see [LS96a] and also Section 2.6 of this chapter), are believed to provide a compelling reason for preferring APES to Capon.

2.2 The Filterbank Approach

Let \{x(t) ; \ t = 1, 2, \ldots, N\} denote the available sample of the stationary signal the spectrum of which is to be estimated, where \(N\) denotes the number of data samples. The filterbank approach basically reduces the problem of estimating the spectrum of \(x(t)\) to that of estimating the amplitude (or power) of a sinusoidal signal buried in colored noise (see, e.g., [LS96a, LSGM86]). More exactly, in the aforementioned approach \(x(t)\) is additively decomposed as

\[ x(t) = \alpha_\omega e^{j\omega t} + \epsilon(t), \quad t = 1, 2, \ldots, N, \quad \omega \in (0, 2\pi] \]  

(2.1)

where \(\alpha_\omega\) denotes the (complex-valued) amplitude of the sinusoidal signal referred to above and \(\epsilon(t)\) is the noise (or residual) term, assumed to be zero-mean. The three main steps of any filterbank spectral estimator can be succinctly described as follows

Step (a): Pass the data \(\{x(t)\}\) through a bandpass filter with (varying) center frequency \(\omega\), to enhance the sinusoidal component in (2.1).

Step (b): Estimate \(\alpha_\omega\) from the filtered data. Let \(\hat{\alpha}_\omega\) denote the estimated amplitude.
Step (c): Take \( \hat{\alpha}_\omega \), for \( \omega \in (0, 2\pi] \), as an estimate of the complex spectrum of \( x(t) \); or \( |\hat{\alpha}_\omega|^2 \) (appropriately normalized) as an estimate of the power spectrum.

We assume that the filter used in Step (a) has a finite impulse response (FIR). Let

\[
h_\omega = [ h_1 \ldots h_L ]^* \tag{2.2}
\]
denote the vector of FIR coefficients, where \( L \) denotes the filter's length and the superscript * stands for the conjugate transpose operator. Observe that the dependence of the vector in (2.2) on the center frequency \( \omega \) is stressed by notation. By using (2.2) we can write the filter output as

\[
z(t) = h_\omega^* y_t, \tag{2.3}
\]
where

\[
y_t = [ x(t) \ldots x(t + L - 1) ]^T \quad (t = 1, \ldots, M). \tag{2.4}
\]

Hereafter, the superscript \( T \) denotes the transpose and

\[
M = N - L + 1. \tag{2.5}
\]

The choice of \( L \) above should be done by a compromise between resolution and statistical stability: the larger \( L \) the better the resolution but the worse the statistical stability. It is not difficult to understand why this is so. The time-bandwidth product type of result implies that the filter can be made narrower in the frequency domain, and hence the corresponding spectral resolution enhanced, as \( L \) increases. On the other hand, \( M \) decreases with increasing \( L \) and therefore we have fewer samples for the estimation of \( \alpha_\omega \) in Step (b), which should lead to a poorer statistical accuracy.

Next we discuss the choice of \( h_\omega \). In the Capon method \( h_\omega \) is chosen such that the power of the filter output is minimized subject to the constraint that the frequency \( \omega \) is passed undistorted

\[
\min_{h_\omega} h_\omega^* R h_\omega \quad \text{subject to} \quad h_\omega^* a_\omega = 1 \tag{2.6}
\]
where

\[
a_\omega = [ 1 \quad e^{i\omega} \ldots e^{i(L-1)\omega} ]^T \tag{2.7}
\]
and

\[
R = E [ y_t y_t^* ]. \tag{2.8}
\]
Hereafter, \( E[\cdot] \) denotes the statistical expectation operator. The solution to the filter design problem in (2.6) is given by [Cap69, Lac71]

\[
h_\omega = \frac{R^{-1} a_\omega}{a_\omega^* R^{-1} a_\omega} \tag{2.9}
\]
assuming that $\mathbf{R}$ is invertible. Of course, (2.9) is not directly implementable since $\mathbf{R}$ is not available. The most commonly used implementable form is obtained by using the sample covariance matrix,

$$\hat{\mathbf{R}} = \frac{1}{M} \sum_{t=1}^{M} \mathbf{y}_t \mathbf{y}_t^*$$

(2.10)

in lieu of $\mathbf{R}$ in (2.9)

$$h_{\omega}^{\text{Capon}} = \frac{\hat{\mathbf{R}}^{-1}\mathbf{a}_\omega}{\text{Tr} \hat{\mathbf{R}}^{-1}\mathbf{a}_\omega \mathbf{a}_\omega^*}. \quad (2.11)$$

The Fourier (see, e.g., [Kay88, Mar87, SM97]) and the APES [LS96a] spectral estimators have not been derived in the filterbank approach framework. However, they can be cast into that framework by simple manipulations. The corresponding FIR filter vectors turn out to be (see, [Kay88, Mar87] and [LS96a], respectively)

$$h_{\omega}^{\text{Fourier}} = \frac{\mathbf{a}_\omega}{L} \quad (2.12)$$

and

$$h_{\omega}^{\text{APES}} = \frac{\hat{\mathbf{Q}}_{\omega}^{-1}\mathbf{a}_\omega}{\text{Tr} \hat{\mathbf{Q}}_{\omega}^{-1}\mathbf{a}_\omega \mathbf{a}_\omega^*} \quad (2.13)$$

where

$$\hat{\mathbf{Q}}_{\omega} = \hat{\mathbf{R}} - \mathbf{Y}_{\omega} \mathbf{Y}_{\omega}^* \quad (2.14)$$

and where

$$\mathbf{Y}_{\omega} = \frac{1}{M} \sum_{t=1}^{M} \mathbf{y}_t e^{-i\omega t} \quad (2.15)$$

In the next section we propose the use of the matched filter (MAFI) approach to derive the filter in Step (a) of the filterbank-based spectral estimation methodology outlined above. Even though the matched filter should be a natural choice for Step (a), apparently its use in the previous context was not directly suggested before.

### 2.3 The Matched-Filterbank Approach

By making use of the notation

$$\mathbf{n}_t = [ \varepsilon(t) \ldots \varepsilon(t + L - 1)]^T \quad (2.16)$$

and

$$\mathbf{s}_t = \alpha_{\omega} e^{i\omega t} \mathbf{a}_\omega \quad (2.17)$$
we can write $y_t$ in (2.4) as

$$y_t = s_t + n_t.$$  

(2.18)

We assume that the initial phase of the sinusoidal signal in the equation above is a random variable which is uniformly distributed over the interval $(0, 2\pi]$ and is independent of the noise term. Then it follows that the covariance matrix of the signal term in (2.18) is given by

$$E [s_t s_t^*] = |a_\omega|^2 a_\omega^* a_\omega^*.$$  

(2.19)

Let $Q_\omega$ denote the covariance matrix (assumed to be invertible) of the noise term in Equation (2.18)

$$Q_\omega = E [n_t n_t^*].$$  

(2.20)

The notation $Q_\omega$ is meant to indicate the fact that the noise (or residual) term in (2.18), and hence its covariance, depends on $\omega$. By definition the matched filter should be designed such that the corresponding signal-to-noise ratio (SNR) in the filter’s output is maximized, i.e.,

$$\max_{h_\omega} \frac{|h_\omega^* a_\omega|^2}{h_\omega^* Q_\omega h_\omega}. $$  

(2.21)

The solution to the design problem above is well known [Kay88, Mar87, SM97] (for completeness we provide a simplest derivation in Appendix 2.A)

$$h_\omega = \frac{Q_\omega^{-1} a_\omega}{a_\omega^* Q_\omega^{-1} a_\omega}. $$  

(2.22)

The covariance matrix in (2.22) can be estimated in several ways. As shown below two natural estimators of $Q_\omega$ lead to the Capon and APES filters, respectively. A third estimate of $Q_\omega$ yields a new filter that is different from both the Capon and APES filters. However, as shown in the next section, the spectral estimator corresponding to the new filter turns out to be equivalent to APES as well.

### 2.3.1 Capon

Since

$$R = |a_\omega|^2 a_\omega^* a_\omega^* + Q_\omega,$$  

(2.23)

we can estimate $Q_\omega$ as

$$\hat{Q}_\omega = \hat{R} - |\hat{a}_\omega|^2 a_\omega^* a_\omega^*.$$  

(2.24)
where \( \hat{\alpha}_\omega \) is an estimate of \( \alpha_\omega \). As we show below the second term in (2.24) has no influence on the \( h_\omega \) in (2.22). This means that for the \( \hat{Q}_\omega \) in (2.24) there will be in fact no need to estimate \( \alpha_\omega \). A straightforward use of the matrix inversion lemma (stated in Appendix 2.B for easy reference) yields

\[
(\hat{R} - |\hat{\alpha}_\omega|^2 a_\omega a_\omega^*)^{-1} a_\omega = \hat{R}^{-1} a_\omega + \frac{|\hat{\alpha}_\omega|^2 \hat{R}^{-1} a_\omega a_\omega^* \hat{R}^{-1} a_\omega}{1 - |\hat{\alpha}_\omega|^2 a_\omega a_\omega^* \hat{R}^{-1} a_\omega} = \frac{\hat{R}^{-1} a_\omega}{1 - |\hat{\alpha}_\omega|^2 a_\omega a_\omega^* \hat{R}^{-1} a_\omega}.
\]

(2.25)

Since the scaling of \( \hat{Q}_\omega^{-1} a_\omega \) in the expression of the sample counterpart of \( h_\omega \) in (2.22) leaves the FIR vector unchanged, it follows from (2.25) that the estimated matched filter corresponding to the \( \hat{Q}_\omega \) in (2.24) coincides with the Capon filter.

A perhaps even more direct way to see that the Capon filter is a matched filter is as follows. Consider the MAFI design problem in (2.21). As the multiplication of \( h_\omega \) by any complex number leaves the ratio in (2.21) unchanged, it follows that the optimization problem in (2.21) is equivalent to

\[
\min_{h_\omega} h_\omega^* Q_\omega h_\omega \quad \text{subject to} \quad h_\omega^* a_\omega = 1.
\]

(2.26)

However, by using (2.23) along with the constraint in (2.26), we can write

\[
h_\omega^* Q_\omega h_\omega = h_\omega^* \left[ R - |\alpha_\omega|^2 a_\omega a_\omega^* \right] h_\omega = h_\omega^* R h_\omega - |\alpha_\omega|^2
\]

(2.27)

The conclusion is that the Capon design problem is nothing but a matched-filter design.

### 2.3.2 APES

By combining (2.17) and (2.18) we obtain

\[
y_t = [\alpha_\omega a_\omega] e^{i\omega t} + n_t.
\]

(2.28)

The least-squares (LS) estimate of the vector \( \alpha_\omega a_\omega \) in (2.28), which ignores the fact that \( a_\omega \) is known, is given by the normalized Fourier transform

\[
[\hat{\alpha}_\omega a_\omega] = \frac{1}{M} \sum_{t=1}^{M} y_t e^{-i\omega t} \triangleq Y_\omega.
\]

(2.29)

Inserting (2.29) into (2.24) yields the following estimate of \( Q_\omega \)

\[
\hat{Q}_\omega = \hat{R} - Y_\omega Y_\omega^*,
\]

(3.30)

the use of which leads to the APES filter in (2.13), (2.14).
2.3.3 MAFI

Equation (2.28) suggests another way to estimate the noise covariance matrix $Q_\omega$

$$
\hat{Q}_\omega = \frac{1}{M} \sum_{t=1}^{M} [y_t - \hat{\alpha}_\omega a_\omega e^{i\omega t}] [y_t - \hat{\alpha}_\omega a_\omega e^{i\omega t}]^*,
$$

(2.31)

where, as before, $\hat{\alpha}_\omega$ denotes an estimate of $\alpha_\omega$. A simple calculation shows that the previous $Q_\omega$ matrix can be rewritten as follows

$$
\hat{Q}_\omega = \hat{R} - \hat{\alpha}_\omega^* Y_\omega a_\omega^* - \hat{\alpha}_\omega a_\omega Y_\omega^* + |\hat{\alpha}_\omega|^2 a_\omega a_\omega^*.
$$

(2.32)

By a calculation similar to (2.25) one can verify that the last term in (2.32) can be dropped when evaluating the MAFI vector. Below we make use of the matrix inversion lemma once more to show that also the third term in (2.32) can be omitted. Let $\Omega$ be the matrix made from the first two terms in (2.32). Then, by the matrix inversion lemma,

$$(\Omega - \hat{\alpha}_\omega a_\omega Y_\omega^*)^{-1} a_\omega = \left[ \Omega^{-1} + \hat{\alpha}_\omega \Omega^{-1} a_\omega Y_\omega^* \Omega^{-1} \right] a_\omega
$$

$$
+ \frac{\Omega^{-1} a_\omega}{1 - \hat{\alpha}_\omega Y_\omega^* \Omega^{-1} a_\omega},
$$

(2.33)

where the dependence on $\omega$ has been omitted to simplify the notation. A further use of the matrix inversion lemma yields

$$
\Omega^{-1} a_\omega = (\hat{R} - \hat{\alpha}_\omega^* Y_\omega a_\omega^*)^{-1} a_\omega
$$

$$
= \left[ \hat{R}^{-1} + \hat{\alpha}_\omega \hat{R}^{-1} Y_\omega a_\omega^* \hat{R}^{-1} \right] a_\omega
$$

$$
= \frac{\hat{R}^{-1} a_\omega - \hat{\alpha}_\omega (a_\omega^* \hat{R}^{-1} Y_\omega) \hat{R}^{-1} a_\omega + \hat{\alpha}_\omega (a_\omega^* \hat{R}^{-1} a_\omega) \hat{R}^{-1} Y_\omega}{1 - \hat{\alpha}_\omega^* a_\omega \hat{R}^{-1} Y_\omega},
$$

which gives the following expression for the MAFI vector

$$
h_{M,AFI} = \frac{\hat{R}^{-1} a_\omega + \hat{\alpha}_\omega [ (a_\omega^* \hat{R}^{-1} a_\omega) \hat{R}^{-1} Y_\omega - (a_\omega^* \hat{R}^{-1} Y_\omega) \hat{R}^{-1} a_\omega ]}{a_\omega^* \hat{R}^{-1} a_\omega}.
$$

(2.34)

The previous filter is generally different from both the Capon and the APES filters. In effect, neither of the latter two depends on an estimate of $\alpha_\omega$, unlike the former. (The estimation of $\alpha_\omega$, needed by (2.34), is discussed in the next section). Despite this fact, in the next section we show the somewhat surprising result that a most reasonable implementation of the spectral estimator corresponding to (2.34) is identical to APES!
2.4 Spectrum Estimation

It is readily checked that all filter vectors previously discussed satisfy

\[ h^*_\omega a_\omega = 1. \]  

By making use of this observation and of (2.28) we obtain

\[ h^*_\omega y_t = \alpha_\omega e^{i\omega t} + h^*_\omega n_t. \]  

The LS estimator of \( \alpha_\omega \) in the above equation is given by

\[ \hat{\alpha}_\omega = h^*_\omega \frac{1}{M} \sum_{t=1}^{M} y_t e^{-i\omega t} = h^*_\omega Y_\omega. \]  

(2.37)

The Capon and APES estimates of the complex spectrum are obtained by inserting the corresponding filter vector expressions in (2.37).

The use of the MAFI filter derived in the previous section for spectrum estimation requires an initial estimate of \( \alpha_\omega \). That estimate might be obtained by any of the other methods discussed herein (i.e., Fourier, Capon or APES), but proceeding in that way would be computationally unacceptable. Actually, we can avoid the need for an initial estimate of \( \alpha_\omega \) in the following way. Inserting (2.34) into (2.37) yields

\[ \hat{\alpha}_\omega (a^*_\omega \hat{R}^{-1} a_\omega) = a^*_\omega \hat{R}^{-1} Y_\omega + \hat{\alpha}_\omega \left[ (a^*_\omega \hat{R}^{-1} a_\omega)(Y^*_\omega \hat{R}^{-1} Y_\omega) - \left| a^*_\omega \hat{R}^{-1} Y_\omega \right|^2 \right] \]

from which we obtain (by assuming that the \( \hat{\alpha}_\omega \)'s in the right-hand and left-hand sides are identical)

\[ \hat{\alpha}_{MAFI} = \frac{a^*_\omega \hat{R}^{-1} Y_\omega}{a^*_\omega \hat{R}^{-1} a_\omega - \left[ (a^*_\omega \hat{R}^{-1} a_\omega)(Y^*_\omega \hat{R}^{-1} Y_\omega) - \left| a^*_\omega \hat{R}^{-1} Y_\omega \right|^2 \right]} \]  

(2.38)

Next we show that the previous spectral estimator coincides with APES

\[ \hat{\alpha}_{APES} = \hat{\alpha}_{MAFI}. \]  

(2.39)

The APES estimate of the complex spectrum is given by (once again we omit the dependence on \( \omega \) for notational convenience)

\[ \hat{\alpha}_{APES} = \frac{a^*_\omega (\hat{R} - Y_\omega Y^*_\omega)^{-1} Y_\omega}{a^*_\omega (\hat{R} - Y_\omega Y^*_\omega)^{-1} a_\omega} \]
\[
\alpha^*_\omega \left( R^{-1} + \frac{\hat{R}^{-1} Y_\omega Y_\omega^* \hat{R}^{-1}}{1 - Y_\omega^* R^{-1} Y_\omega} \right) Y_\omega \\
\alpha^*_\omega \left( R^{-1} + \frac{\hat{R}^{-1} Y_\omega Y_\omega^* \hat{R}^{-1}}{1 - Y_\omega^* R^{-1} Y_\omega} \right) a_\omega \\
= \frac{\alpha^*_\omega \hat{R}^{-1} Y_\omega}{\alpha^*_\omega \hat{R}^{-1} a_\omega - \left( \alpha^*_\omega \hat{R}^{-1} a_\omega \right) (Y_\omega^* \hat{R}^{-1} Y_\omega) - \left| \alpha^*_\omega \hat{R}^{-1} Y_\omega \right|^2}
\]

which is identical to (2.38). Hence the equality in (2.39) is proved.

For comparative purposes, note that the (Amplitude Spectrum) Capon\(^2\) spectral estimator is given by

\[
\hat{\alpha}_{\text{Capon}} = \frac{\alpha^*_\omega \hat{R}^{-1} Y_\omega}{\alpha^*_\omega \hat{R}^{-1} a_\omega}
\]

The next section discusses the relative merits of Capon and APES.

2.5 Comparisons and Discussions

2.5.1 The Computational Issue

Let \( \hat{R}^{-1/2} \) denote a square root of the positive definite matrix \( \hat{R}^{-1} \), and let

\[
\nu_\omega = \hat{R}^{-1/2} a_\omega \\
\mu_\omega = \hat{R}^{-1/2} Y_\omega.
\]

The Capon and APES spectral estimators can be expressed as relatively simple functions of the \( \nu_\omega \) and \( \mu_\omega \) introduced above, i.e.,

\[
\hat{\alpha}_{\text{Capon}} = \frac{\nu_\omega^* \mu_\omega}{\nu_\omega^* \nu_\omega} \triangleq \frac{\nu_\omega^* \mu_\omega}{\| \nu_\omega \|^2}
\]

\[
\hat{\alpha}_{\text{APES}} = \frac{\nu_\omega^* \mu_\omega}{\| \nu_\omega \|^2 - \left( \| \nu_\omega \|^2 \| \mu_\omega \|^2 - \| \nu_\omega^* \mu_\omega \|^2 \right)}
\]

The efficient estimation of the Capon spectrum estimate is discussed, for example, in [Kay88, Mar87, SM97]. We do not intend to dwell into the details of that computation here.\(^3\) The point we want to make here is that APES is only slightly more involved computationally than Capon (as should be evident by comparing (2.44) and (2.45)), and hence the efficient computational means developed for the latter can also be applied to the former.

\(^2\)See also the discussion in Section 1.2.2.

\(^3\)A further discussion on this issue is found in Chapter 3.
2.5.2 The Statistical Performance Issue

The Capon, APES as well as the Fourier spectral estimators can be shown to have the same asymptotic variance under the following condition

\[ C: \text{The signal } x(t) \text{ can be written as in (2.1), where } \epsilon(t) \text{ is a zero-mean stationary random process with finite spectral density at } \omega \]

\[ \phi_\epsilon(\omega) < \infty \]  

(2.46)

In more exact terms, the following result holds true.

**Theorem 2.1** Under condition \( C \) and the additional assumption that \( \epsilon(t) \) is circularly symmetrically distributed, the estimation errors in the Fourier, Capon and APES spectral estimators are asymptotically circularly symmetrically distributed with zero-mean and the following common variance

\[ \lim_{M \to \infty} ME \left[ |\hat{\alpha}_\omega - \alpha_\omega|^2 \right] = \phi_\epsilon(\omega) \]  

(2.47)

**Proof:** See Appendix 2.C.

The need to enforce condition \( C \) limits, to some extent, the importance of the previous result. Indeed the assumption made in \( C \) is satisfied if (and essentially only if) the signal \( x(t) \) has a mixed spectrum and \( \omega \) is the location of a spectral line. In some applications, such as target feature extraction by means of a Doppler radar, the signal has a mixed spectrum and estimating the amplitudes of the sinusoidal components is a key problem (see, e.g., [LS96a] and the references therein). The result of Theorem 2.1 is relevant to such applications. In other applications, however, the main interest is in the continuous component of the spectrum. Condition \( C \) does not hold exactly if the spectrum is continuous at \( \omega \).

The previous result is of a somewhat limited interest also because of its asymptotic character. Indeed, in simulations with medium or small-sized samples, the spectral estimators under study have been found to behave quite differently from one another in contradiction with what is predicted by the (asymptotic) result of Theorem 2.1 (see, e.g., [LS96a]; also see the next section). In particular, in scenarios with multiple, closely spaced spectral lines the Fourier estimate of the complex spectrum has been found to be (sometimes significantly) less accurate than the spectrum estimates obtained by using the Capon and APES methods [LS96a, LSGM86] (see also the resolution comparison in Section 3.4).
The finite-sample analysis of the spectral estimators under discussion would be of considerable interest. However, while this is possible for the Fourier estimator, a complete analysis of the more interesting, data-adaptive Capon and APES estimators appears to be quite difficult at best. A partial analysis of the latter estimators is nevertheless possible as shown in what follows. Specifically, in Appendix 2.D we make use of a higher-order Taylor series expansion technique to prove that, to within a second-order approximation, and under the additional mild assumption that the third-order moments of $n_t$ are zero, Capon is biased downwards whereas APES is unbiased

$$\frac{ME[\hat{\alpha}_{\text{Capon}} - \alpha_\omega]}{\alpha_\omega} < 0, \quad \omega \in [-\pi, \pi)$$  \hspace{1cm} (2.48)

and

$$\frac{ME[\hat{\alpha}_{\text{APES}} - \alpha_\omega]}{\alpha_\omega} = 0, \quad \omega \in [-\pi, \pi)$$  \hspace{1cm} (2.49)

for sufficiently large values of $M$.

The above result provides theoretical support to the empirically observed fact that Capon underestimates the complex spectrum, whereas the APES estimate is almost unbiased (see [LS96a]). In fact the Capon and APES spectral estimators satisfy the following neat inequality, which points to the type of result proved above

$$|\hat{\alpha}_{\text{Capon}}| \leq |\hat{\alpha}_{\text{APES}}| \quad \text{and} \quad \arg(\hat{\alpha}_{\text{Capon}}) = \arg(\hat{\alpha}_{\text{APES}}), \quad \forall \omega \in [-\pi, \pi)$$  \hspace{1cm} (2.50)

where the modulus equality holds only for $M \to \infty$. To check (2.50) observe that the term between square parentheses in (2.40) is positive (by the Cauchy-Schwartz inequality) and that it can be zero if and only if $Y_\omega$ is proportional to $a_\omega$ (which can hold true only when $M \to \infty$).

We believe that (2.50) along with (2.48) and (2.49) provide a theoretical motivation for preferring APES to Capon in most spectrum estimation exercises. The simulations in the next section lend further evidence to the fact that APES outperforms Capon in the finite-sample case.
2.6 Numerical Examples

In this section we study the accuracy of the APES and Capon amplitude estimates in a number of cases.

![Figure 2.1: The modulus of the true spectrum.](image)

2.6.1 Estimation Performance Versus SNR

We begin by studying the performance of using the APES and Capon methods for complex amplitude estimation as the SNR varies. Figure 2.1 shows the modulus of the true spectrum of the sinusoidal signal, which consists of four dominant spectral lines and nine small spectral lines located at the following frequencies: 0.0625, 0.0875, 0.25, 0.285, 0.33, 0.35, 0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49. The sinusoids have initial phase $\pi/4$. The data sequence has $N = 64$ data samples and is corrupted by complex white Gaussian noise with zero mean and variance $\sigma^2$. The SNR for the $k$th spectral line is defined as

$$\text{SNR}_k = 10 \log_{10} \left( \frac{|\alpha_k|^2}{\sigma^2} \right) \text{ [dB]},$$

(2.51)

where $\alpha_k$ is the complex amplitude of the $k$th spectral line. In the following, we compare the biases and variances of the APES (solid lines in the figures) and Capon (dashed lines in the figures) estimates of the complex amplitude at the frequency of the first dominant spectral line, as its SNR varies (see also Chapter 4). Our results are obtained from 100 Monte Carlo trials.
Figure 2.2: Bias and variance of the complex amplitude estimate at the frequency of the first spectral line as a function of SNR when \( L = 24 \). (a) Real part of the bias. (b) Imaginary part of bias. (c) Variance. (d) MSE.
Figures 2.2(a) and 2.2(b) compare the real and imaginary parts, respectively, of the biases of the Capon and APES estimators, for the first spectral line (the others behave similarly), when \( L = 24 \). As seen from the figures, the APES estimates are nearly unbiased, while the Capon estimates are biased downward, as was expected from our previous theoretical analysis. Figure 2.2(c) and 2.2(d) compare the variances and the MSE, respectively, of the Capon and APES estimators when \( L = 24 \). Note that the APES estimator provides not only a smaller bias but also a lower variance than the Capon estimator.

### 2.6.2 Estimation Performance Versus Filter Length

Next we study the effect of the filter length, \( L \), on the performance of the two estimators. The data studied is the same as in the previous example, with the SNR of the first dominant spectral line fixed at 20 dB. Our results are obtained from 100 Monte Carlo trials.

Figures 2.3(a) and 2.3(b) compare the real and imaginary parts, respectively, of the biases of the Capon and APES estimators, for the first spectral line, as \( L \) varies. As seen from the figures, the APES estimates are again almost unbiased for not too small filter lengths, while the Capon estimates have very large biases for large \( L \). Figure 2.3(c) and 2.3(d) compares the variances and the MSE, respectively, of the Capon and APES estimators. Note that the variance of the APES estimate varies little for the studied filter lengths, while the variance of the Capon estimate significantly increases with increasing \( L \). This type of behavior makes the choice of \( L \) with the Capon method quite difficult. We remind the reader that to enhance the resolution \( L \) should be chosen as large as possible (subject to \( L < N/2 \) so that \( \mathbf{R} \) is not singular). While APES with this choice of \( L \) works fine, Capon does not. We further illustrate this point in the next figure.

In Figures 2.4(a) and 2.4(b), one realization of the modulus of the estimated spectra obtained by using APES and Capon, respectively, is compared with the true spectrum (dotted lines) for \( L = 15 \). Figures 2.4(c) and 2.4(d) are similar to Figures 2.4(a) and 2.4(b) except that the filter length is now increased to \( L = 24 \). As can be seen from the figures, APES gives much more accurate peak amplitude estimates, at the price of slightly wider spectral peaks. Note that, as \( L \) increases, the resolution of both methods improves, as expected; however, unlike APES, the performance of the Capon amplitude estimates deteriorates significantly with increasing \( L \). Note also that the APES estimates, at all evaluated frequencies, are larger than the Capon estimates, as expected.
Figure 2.3: Bias and variance of the complex amplitude estimate at the frequency of the first spectral line as a function of $L$ when the SNR of the first spectral line is 20 dB. (a) Real part of the bias. (b) Imaginary part of bias. (c) Variance. (d) MSE.
Figure 2.4: Modulus of the estimated spectrum (solid lines) compared with the true spectrum (dotted lines) when the SNR of the first spectral line is 20 dB. (a) APES with $L = 15$. (b) Capon with $L = 15$. (c) APES with $L = 24$. (d) Capon with $L = 24$. 
2.7 Conclusions

The Capon spectral estimator has been shown to be a matched-filterbank method. In fact, the Capon's filter design was shown to be precisely a matched filter design. The APES spectral estimator has also been shown to be a member of the class of matched-filterbank methods. In effect, APES was proved to coincide with two seemingly different matched-filterbank spectral estimation methods.

The matched-filterbank interpretation of Capon and APES provides a (partial) theoretical explanation for the good, empirically observed performance of these two spectral estimators. For the moment it is an open issue whether other matched-filterbank spectral estimation methods with enhanced properties exist (see also Section 1.5).

The matched-filterbank interpretation has also provided further insight into the properties of and relationships between Capon and APES. In particular the signal model used by the two methods was shown to be adequate only at the frequencies corresponding to the spectral lines in a mixed spectrum. The model in question is not valid at the frequencies of the continuous spectral component. While APES was shown by means of simulations to yield accurate estimates of the latter component as well [LS96a], there is a clear need for a better theoretical understanding of why and how APES works in such cases (The performance of the Capon estimate of the continuous spectral component was poor in the simulations reported in [LS96a]). Derivation of matched-filterbank methods specifically designed for continuous spectra is also an interesting open issue.

The casting of Capon and APES into the matched-filterbank framework, along with the related analysis, has also led to a neat result concerning the relationship between these two methods: the Capon spectral estimate is always smaller than APES's, for all frequencies. This theoretical observation lends support to the empirical finding that Capon often underestimates the true spectrum whereas APES is nearly unbiased. In fact by making use of an expansion technique we were able to prove that, to within a second-order approximation, Capon is indeed biased downwards whereas APES is unbiased.

As APES is only slightly more involved than Capon, from a computational standpoint, preferring the former to the latter appears to be a logical conclusion of both the theoretical and empirical analysis in this chapter.
2.A Proof of (2.22)

Let \( Q_{\omega}^{1/2} \) denote a Hermitian (for notational simplicity) square root of \( Q_{\omega} \). Then, by the Cauchy-Schwartz inequality,

\[
\frac{\left| h_{\omega}^* Q_{\omega}^{1/2} Q_{\omega}^{-1/2} a_{\omega} \right|^2}{h_{\omega}^* Q_{\omega} h_{\omega}} \leq a_{\omega}^* Q_{\omega}^{-1} a_{\omega},
\]

where the equality is achieved if and only if

\[
h_{\omega} = \beta Q_{\omega}^{-1} a_{\omega}
\]

for any scalar \( \beta \in \mathbb{C} \). As is made transparent in Section 2.4, it is often required that \( h_{\omega} \) passes the frequency \( \omega \) undistorted, that is \( h_{\omega}^* a_{\omega} = 1 \). This requirement can be met by choosing \( \beta \) as \( \beta = 1/ \left[ a_{\omega}^* Q_{\omega}^{-1} a_{\omega} \right] \), which yields (2.22).

2.B The Matrix Inversion Lemma

Let \( R \) be an \( n \times n \) matrix, and let \( b \) and \( c \) be \( n \times 1 \) vectors. Then, under the assumption that the inverse appearing below exists,

\[
(R - bc^*)^{-1} = R^{-1} + \frac{R^{-1}bc^*R^{-1}}{1 - c^*R^{-1}b}
\]

(see, e.g., [Kay88]).
2.C  Proof of Theorem 2.1

To simplify the notation we omit the dependence on \( \omega \) whenever there is
no possibility for confusion. From (2.28) we have that

\[
Y_\omega = \alpha_\omega a_\omega + \Delta
\]  

(2.C.1)

where

\[
\Delta = \frac{1}{M} \sum_{t=1}^{M} n_t e^{-i\omega t}.
\]  

(2.C.2)

The first and second-order moments of \( \Delta \) are easy to calculate

\[
E(\Delta) = 0
\]  

(2.C.3)

\[
E(\Delta \Delta^T) = 0
\]  

(2.C.4)

(by the circularly symmetric distribution assumption) and

\[
E(\Delta \Delta^*) = \frac{1}{M^2} \sum_{t=1}^{M} \sum_{s=1}^{M} R_n(t-s) e^{-i\omega(t-s)}
\]

\[
= \frac{1}{M^2} \sum_{\tau=-M}^{M} (M-|\tau|) R_n(\tau) e^{-i\omega \tau},
\]  

(2.C.5)

where \( \{R_n(\tau)\} \) is the covariance sequence of \( n_t \). It follows from (2.C.5) and
condition C that

\[
\lim_{M \to \infty} ME(\Delta \Delta^*) = \sum_{\tau=-\infty}^{\infty} R_n(\tau) e^{-i\omega \tau} = \phi_n(\omega) = \phi_\epsilon(\omega) a_\omega a_\omega^*,
\]  

(2.C.6)

where the last equality follows from standard results on the transfer of
spectral densities through linear systems. Among others, the previous
calculations imply that, as \( M \to \infty \), \( Y_\omega \) tends to \( \alpha_\omega a_\omega \) (in the mean square
sense). Hence \( h^C_{\text{apon}} \) and \( h^APES_\omega \) have the same limit as \( M \) increases
without bound.

Let \( h \) denote a generic FIR vector and let \( h_\infty \) denote the deterministic
vector that is the limit of (the possibly random) \( h \) when \( M \) goes to infinity.
Observe that for all methods under study the associated \( h \) and \( h_\infty \) vectors satisfy

\[
h^* a_\omega = 1 \quad \text{and} \quad h^*_\infty a_\omega = 1.
\]  

(2.C.7)

By using this observation along with (2.C.1) we obtain

\[
\hat{\alpha}_\omega \overset{\Delta}{=} h^* Y_\omega = \alpha_\omega + h^* \Delta
\]  

(2.C.8)
Because $\Delta$ tends to zero as $M \to \infty$ (as shown previously), it follows from (2.C.8) that the estimation error in $\hat{\alpha}_\omega$ can asymptotically be written as (to within a first-order approximation)

$$\hat{\alpha}_\omega - \alpha_\omega \simeq h_\infty^* \Delta.$$  \hfill (2.C.9)

From (2.C.9) and (2.C.3)-(2.C.6) we readily derive that

$$ME(\hat{\alpha}_\omega - \alpha_\omega)^2 = 0 \quad \text{as } M \to \infty$$

and

$$\lim_{M \to \infty} ME|\hat{\alpha}_\omega - \alpha_\omega|^2 = h_\infty^* \lim_{M \to \infty} ME(\Delta \Delta^*) h_\infty$$

$$= \phi_c(\omega) |h_\infty^* a_\omega|^2$$

$$= \phi_c(\omega)$$

and the proof is concluded.
2.D Proof of (2.48), (2.49)

Once again we omit the dependence on \( \omega \) for notational convenience.

**Proof of (2.48)**

By using (2.9) and (2.C.8) we obtain

\[
\hat{\alpha}_{Capon} - \alpha_\omega = \frac{a_\omega^* \hat{R}^{-1} \Delta}{a_\omega^* R^{-1} a_\omega} \tag{2.D.1}
\]

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 (2.D.1) yields

\[
\hat{\alpha}_{Capon} - \alpha_\omega = \frac{a_\omega^* (\hat{R}^{-1} - R^{-1}) \Delta}{a_\omega^* R^{-1} a_\omega}
+ \frac{a_\omega^* \hat{R}^{-1} (R - \hat{R}) R^{-1} \Delta}{a_\omega^* R^{-1} a_\omega}
+ \frac{1}{a_\omega^* R^{-1} a_\omega} - \frac{1}{a_\omega^* R^{-1} a_\omega} + \frac{1}{a_\omega^* R^{-1} a_\omega}
\]

\[
\approx - \frac{a_\omega^* R^{-1} (R - \hat{R}) R^{-1} \Delta}{a_\omega^* R^{-1} a_\omega}
+ \frac{a_\omega^* R^{-1} \Delta}{(a_\omega^* R^{-1} a_\omega)^2}
\]

which, in turn, implies that

\[
E \left[ \hat{\alpha}_{Capon} - \alpha_\omega \right] \approx
E \left[ \frac{-a_\omega^* \hat{R} R^{-1} \Delta}{a_\omega^* R^{-1} a_\omega} \right]
\]

\[
\left( a_\omega^* R^{-1} a_\omega \right)^2
\]

\[
(2.D.2)
\]

Next we note that

\[
\hat{R} = \frac{1}{M} \sum_{t=1}^{M} [\alpha_\omega a_\omega e^{i\omega t} + n_t] \left[ \alpha_\omega^* a_\omega e^{-i\omega t} + n_t^* \right]
\]
\[
= |\alpha_\omega|^2 a_\omega a_\omega^* + \alpha_\omega^* \Delta a_\omega^* + \alpha_\omega a_\omega \Delta^* + \frac{1}{M} \sum_{t=1}^{M} n_t n_t^*
\]

and also remind the reader of the definition (2.3.2) of \( \Delta \) and the assumption that \( n_t \) has zero third-order moments. By making use of these facts, along with (2.3.2), we can write

\[
(a_\omega^* R^{-1} a_\omega)^2 E (\hat{\alpha}_{Capon} - \alpha_\omega) \approx \\
\approx E \left\{ -(a_\omega^* R^{-1} a_\omega) (a_\omega^* R^{-1} \Delta a_\omega^* R^{-1} \Delta \alpha_\omega + a_\omega^* R^{-1} a_\omega \Delta^* R^{-1} \Delta \alpha_\omega) + \\
+ (a_\omega^* R^{-1} \Delta) (a_\omega^* a_\omega R^{-1} a_\omega + a_\omega a_\omega R^{-1} a_\omega \Delta^* R^{-1} a_\omega) \right\} \\
= E \left\{ -a_\omega^* (a_\omega^* R^{-1} a_\omega)(a_\omega^* R^{-1} \Delta)^2 - \alpha_\omega (a_\omega^* R^{-1} a_\omega)^2 (\Delta^* R^{-1} \Delta) + \\
+ \alpha_\omega^* (a_\omega^* R^{-1} \Delta)^2 (a_\omega^* R^{-1} a_\omega) + \alpha_\omega (a_\omega^* R^{-1} a_\omega) |a_\omega^* R^{-1} \Delta|^2 \right\} \\
= \alpha_\omega (a_\omega^* R^{-1} a_\omega) E \left\{ |a_\omega^* R^{-1} \Delta|^2 - (a_\omega^* R^{-1} a_\omega)(\Delta^* R^{-1} \Delta) \right\}
\]

The quantity between curly parentheses above is negative and thus is its expectation (by the Cauchy-Schwarz inequality). Hence (2.48) follows.

**Proof of (2.49)**

The APES estimate of the complex spectrum is given by (c.f. (2.39))

\[
\hat{\alpha}_{APES} = \frac{a_\omega^* \hat{R}^{-1} Y_\omega}{a_\omega^* \hat{R}^{-1} a_\omega - \psi} \quad (2.4.3)
\]

where

\[
\psi = (a_\omega^* \hat{R}^{-1} a_\omega)(Y_\omega^* \hat{R}^{-1} Y_\omega) - |a_\omega^* \hat{R}^{-1} Y_\omega|^2. \quad (2.4.4)
\]

By using (2.3.1) in (2.4.4) we obtain

\[
\psi = (a_\omega^* \hat{R}^{-1} a_\omega) \left( |\alpha_\omega|^2 a_\omega^* R^{-1} a_\omega + \alpha_\omega a_\omega^* \hat{R}^{-1} a_\omega + \alpha_\omega \Delta^* \hat{R}^{-1} a_\omega + \Delta a_\omega^* \hat{R}^{-1} \Delta a_\omega \right) - \\
- |\alpha_\omega|^2 (a_\omega^* \hat{R}^{-1} a_\omega)^2 - \alpha_\omega (a_\omega^* \hat{R}^{-1} a_\omega)(\Delta^* \hat{R}^{-1} a_\omega) - \\
- \alpha_\omega^* (a_\omega^* \hat{R}^{-1} \Delta)(a_\omega^* \hat{R}^{-1} a_\omega) - |a_\omega^* \hat{R}^{-1} \Delta|^2 \\
= (a_\omega^* \hat{R}^{-1} a_\omega)(\Delta^* \hat{R}^{-1} \Delta) - |a_\omega^* \hat{R}^{-1} \Delta|^2 \\
\approx (a_\omega^* \hat{R}^{-1} a_\omega)(\Delta^* \hat{R}^{-1} \Delta) - |a_\omega^* \hat{R}^{-1} \Delta|^2 \triangleq \overline{\psi}.
\]

Next note that

\[
\frac{1}{a_\omega^* \hat{R}^{-1} a_\omega - \psi} = \frac{1}{a_\omega^* \hat{R}^{-1} a_\omega} + \frac{1}{a_\omega^* \hat{R}^{-1} a_\omega - \psi} - \frac{1}{a_\omega^* \hat{R}^{-1} a_\omega} \\
= \frac{1}{a_\omega^* \hat{R}^{-1} a_\omega} + \frac{\psi}{(a_\omega^* \hat{R}^{-1} a_\omega)(a_\omega^* \hat{R}^{-1} a_\omega - \psi)}
\]
Consequently,
\[
\hat{\alpha}_{APES} - \alpha_\omega = \frac{a_\omega^* \hat{R}^{-1} (a_\omega \alpha - \Delta) - \alpha_\omega (a_\omega^* \hat{R}^{-1} a_\omega) + \alpha_\omega \psi}{a_\omega^* \hat{R}^{-1} a_\omega - \psi}
\]
\[
= \frac{a_\omega^* \hat{R}^{-1} \Delta + \alpha_\omega \psi}{a_\omega^* \hat{R}^{-1} a_\omega - \psi}
\]
\[
= \frac{a_\omega^* \hat{R}^{-1} \Delta + \alpha_\omega \psi}{a_\omega^* \hat{R}^{-1} a_\omega} + \frac{\psi(\alpha_\omega^* \hat{R}^{-1} \Delta + \alpha_\omega \psi)}{(a_\omega^* \hat{R}^{-1} a_\omega) (a_\omega^* \hat{R}^{-1} a_\omega - \psi)}
\]
\[
\approx \frac{a_\omega^* \hat{R}^{-1} \Delta}{a_\omega^* \hat{R}^{-1} a_\omega} + \frac{\alpha_\omega \psi}{a_\omega^* \hat{R}^{-1} a_\omega}.
\] (2.D.5)

The first term in (2.D.5) is equal to \((\hat{\alpha}_{Capon} - \alpha_\omega)\) and hence it follows from the first part of this appendix that the expectation of this term satisfies (to within a second-order approximation)
\[
E \left[ \frac{a_\omega^* \hat{R}^{-1} \Delta}{a_\omega^* \hat{R}^{-1} a_\omega} \right] \approx -\alpha_\omega E(\psi) a_\omega^* \hat{R}^{-1} a_\omega.
\] (2.D.6)

Combining (2.D.5) and (2.D.6) yields
\[
E(\hat{\alpha}_{APES} - \alpha_\omega) \approx 0
\]
and the proof is almost concluded. It only remains to motivate the normalizing factor \(M\) used in both (2.48) and (2.49). However the factor in question follows relatively easily from the fact that \(\Delta\) is \(\mathcal{O}(1/M^{1/2})\) (in the mean square sense) and this implies that the second-order approximations previously used for both \(E(\hat{\alpha}_{Capon} - \alpha_\omega)\) and \(E(\hat{\alpha}_{APES} - \alpha_\omega)\) are \(\mathcal{O}(1/M)\).
Chapter 3

On Efficient Implementation of the Capon Algorithm

We present an efficient implementation of the Amplitude Spectrum Capon (ASC) estimator, denoted the Burg-based ASC (BASC) estimator. The implementation, which is based on the Burg algorithm and the Fast Fourier Transform (FFT), will be significantly faster than the implementation recently proposed by Liu et al. The BASC implementation can be seen as a combination of the algorithm by Liu et al and the efficient Power Spectrum Capon (PSC) algorithm proposed by Musicus.

3.1 Introduction

There has of lately been an increased interest in non-parametric spectral estimation, in particular using the Power Spectrum Capon (PSC) and the Amplitude Spectrum Capon (ASC) spectral estimators, as well as the recently proposed APES (Amplitude and Phase ESTimation) spectral estimator [LS96a]. In Chapter 2, it was found that APES, as well as the PSC and the ASC methods, could be interpreted as members of the class of matched-filterbank (MAFI) spectral estimators. With the MAFI interpretation, the APES estimator was given a simpler and more intuitive derivation, and could be seen as a Capon-like method using an estimate of the noise covariance matrix. This fact much simplifies the comparison between the estimators.

An important difference between the estimators is that ASC and APES estimate the amplitude and phase of a complex-valued sinusoid located at the frequency of interest, whereas the PSC estimates the power of the filter output. Thus, the PSC estimator does not yield an estimate of the signal's
phase information, which the APES and the ASC estimators do.

In this chapter we present a computationally efficient implementation of the ASC spectral estimation algorithm. The algorithm is based on the famous Burg algorithm \cite{Bur75} (see also, e.g., \cite{SM97}), and is denoted the Burg-based ASC (BASC). The proposed implementation, which will depend only on the (forward) linear prediction coefficients (LPCs) and the prediction error autocovariance sequence, can be implemented using the Fast Fourier Transform (FFT). The algorithm has close connections to both the efficient ASC implementation recently presented by Liu et al \cite{LLL98} and the efficient PSC implementation presented by Musicus \cite{Mus85}. The algorithm in \cite{LLL98} is based on the results presented in Chapter 2 (see also \cite{LSLJ97, LLS98}). On the other hand, the Musicus algorithm is based on the closed form inverse of a Toeplitz matrix, depending only on the LPCs as computed by the Levinson-Durbin algorithm. We propose a combination of the two, computing the in \cite{LLL98} needed Cholesky factor of the sample covariance matrix using the LPCs. It is well known that the autoregressive (AR) spectral estimates obtained by using the Burg algorithm will have higher resolution than the AR spectral estimates obtained from the solution to the Yule-Walker equations \cite{UB75}. The same will also hold for the Capon estimators, and we will thus make use of the Burg algorithm, appropriately weighted, to compute the LPCs in our proposed implementation.

The chapter is organized as follows: In the next section the MAFI interpretation of the filterbank spectral estimators is reviewed. In Section 3.3, the computationally efficient BASC algorithm is presented. In Section 3.4, we illustrate and compare the performance of the different estimators. Finally, Section 3.5 contains our conclusions.

### 3.2 The Matched Filterbank Approach

In the filterbank approach to spectral estimation, the spectrum is estimated by passing the signal through an $L$-tap narrowband filter, $h_\omega$, with varying center frequency $\omega$ (see, e.g., \cite{LS96a, SM97}). Here, and in the following, the subscript $\omega$ is used to indicate a dependence on the filter’s center frequency.

Let $\{x(t); t = 0, \ldots, N - 1\}$ denote the available (stationary) data sample of which the spectrum is to be estimated. The filter output can then be written as

$$h_\omega^T y(t) = \alpha_\omega e^{i\omega t} + \epsilon(t), \quad t = 0, \ldots, M - 1 \quad (3.2.1)$$

where $(\cdot)^T$ and $(\cdot)^*$ denote transpose and complex conjugate transpose,
\( M = N - L + 1, \)
\[
\mathbf{y}(t) = \begin{bmatrix} x(t) & \ldots & x(t + L - 1) \end{bmatrix}^T,
\] (3.2.2)
and where \( \epsilon(t) \) is some additive colored noise. In the following we will assume that \( \epsilon(t) \) is uncorrelated with \( \alpha(\omega)e^{i\omega t} \). This assumption is basically only true for spectral lines, although it has been found to be a reasonable approximation in other cases as well [LS96a].

The least-squares estimate of the complex amplitude, \( \alpha(\omega) \), in (3.2.1) is given by
\[
\hat{\alpha}_\omega = \mathbf{h}_\omega^* \mathbf{y}_\omega,
\] (3.2.3)
where
\[
\mathbf{y}_\omega = \frac{1}{M} \sum_{t=0}^{M-1} \mathbf{y}(t) e^{-i\omega t}.
\] (3.2.4)

The problem of designing the filter \( \mathbf{h}_\omega \) as a matched-filterbank (MAFI) was studied in Chapter 2. It was found that the Capon method can be interpreted as a member of the MAFI class. The corresponding filter is given by
\[
\mathbf{h}_\omega = \frac{\hat{\mathbf{R}}^{-1} \mathbf{a}_L(\omega)}{\mathbf{a}_L(\omega)^* \hat{\mathbf{R}}^{-1} \mathbf{a}_L(\omega)}
\] (3.2.5)
where
\[
\mathbf{a}_L(\omega) = \begin{bmatrix} 1 & e^{i\omega} & \ldots & e^{i(L-1)\omega} \end{bmatrix}^T,
\] (3.2.6)
and where \( \hat{\mathbf{R}} \) is an estimate of the sample covariance matrix
\[
\mathbf{R} = E \{ \mathbf{y}(t) \mathbf{y}(t)^* \}.
\] (3.2.7)

Here, \( E \{ \cdot \} \) denotes expectation. We will in our numerical examples study the forward-only (F) sample covariance matrix estimate
\[
\hat{\mathbf{R}}^F = \frac{1}{M} \sum_{t=0}^{M-1} \mathbf{y}(t) \mathbf{y}(t)^* \],
\] (3.2.8)
and the forward-backward averaged (FB) sample covariance matrix estimate
\[
\hat{\mathbf{R}}^{FB} = \frac{1}{2} \left( \hat{\mathbf{R}}^F + \mathbf{J} \left( \hat{\mathbf{R}}^F \right)^T \mathbf{J} \right)
\] (3.2.9)
(see [JS99] for an analysis and a comparison between the use of the forward-only and the forward-backward sample covariance matrices for the Capon
spectral estimator). Here, $J$ denotes the so-called exchange matrix, whose anti-diagonal elements are ones and all other elements are zero.

The ASC amplitude estimate, at frequency $\omega$, is given by (3.2.3) evaluated using the filter (3.2.5), i.e.,

$$\hat{\alpha}_\omega = \frac{a_L(\omega) \hat{R}^{-1} Y_\omega}{a_L(\omega)^* R^{-1} a_L(\omega)}; \quad (3.2.10)$$

and the corresponding amplitude spectrum estimate is given as the magnitude square of (3.2.10), i.e.,

$$\hat{\varphi}_\omega = |\hat{\alpha}_\omega|^2. \quad (3.2.11)$$

Note that ASC will in general yield a different, and often preferable, spectral estimate than the PSC spectral estimator, which is obtained by estimating the power of the filter output, i.e.,

$$\hat{\varphi}_\omega = \frac{1}{M} \sum_{t=0}^{M-1} |h^*_\omega y(t)|^2 = h^*_\omega \hat{R} h_\omega = \frac{1}{a_L^* R^{-1} a_\omega}; \quad (3.2.12)$$

(see also the discussion in Section 1.2.3).

### 3.3 Proposed Efficient Implementation

In this section we propose an efficient implementation of the ASC spectral estimator. The implementation is based on the efficient computation of the Cholesky factors of the inverse covariance matrix using the Burg algorithm. Let

$$C \triangleq R^{-1/2} \quad (3.3.13)$$

denote a square root of the positive definite matrix $R^{-1}$, and let (cf. (2.42) and (2.43))

$$\nu_\omega = C^* a_L(\omega) \quad (3.3.14)$$

$$\mu_\omega = C^* Y_\omega$$

$$= \frac{1}{M} C^* W a_M(\omega), \quad (3.3.15)$$

where $x$ denotes the complex conjugate of $x$, and

$$W = [ y(1) \ldots y(M) ]. \quad (3.3.16)$$
Note that, the vectors $\nu_\omega$ and $\mu_\omega$ can be efficiently calculated for all frequencies on the frequency grid using the FFT. A column wise FFT of $C$ gives the $\nu_\omega^\ast$, whereas a row wise FFT of $C^\ast W/M$ gives $\mu_\omega$ for all the frequencies considered.

Making use of (3.2.12), as well as (3.3.14), the PSC spectral estimate can be formulated as

$$\varphi^{\text{PSC}}_\omega = \frac{1}{\nu_\omega^\ast \nu_\omega}.$$  \hfill (3.3.17)

Similarly, the ASC spectral estimate in (3.2.11) can be found as (cf. (2.44))

$$\varphi^{\text{ASC}}_\omega = \left| \frac{\nu_\omega^\ast \mu_\omega}{\nu_\omega^\ast \nu_\omega} \right|^2.$$  \hfill (3.3.18)

An efficient computation of (3.3.18) was recently proposed in [LLL98]. There, $C$ was estimated by computing the Cholesky factorization of the outer-product sample covariance estimate $\hat{R}$. This approach requires the computing of $\hat{R}$ as well as the Cholesky factor $\hat{C}$. Here, we instead propose to compute the ASC estimate in (3.3.18) by constructing $C$ from the forward LPCs (see, e.g., [SS89], Complement C8.2)

$$\hat{C} = \begin{bmatrix} 1 & 0 \\ \vdots & \ddots & \ddots \\ a_{L-1,L-1} & \cdots & a_{1,1} & 1 \end{bmatrix} \begin{bmatrix} \sigma_{L-1}^{-1} & \cdots & 0 \\ 0 & \sigma_{L-2}^{-1} & \cdots \\ \vdots & \ddots & \ddots \\ 0 & \cdots & 0 & \sigma_0^{-1} \end{bmatrix}$$  \hfill (3.3.19)

where $\{a_{k,n}\}$ and $\sigma_n$ denote the coefficients and the standard deviation of the (forward) linear prediction model of order $n$, satisfying

$$e_n(t) = x(t) + \sum_{k=1}^{n} a_{k,n} x(t-k).$$  \hfill (3.3.20)

Here, $\sigma_n$ is the standard deviation of the prediction error $e_n(t)$. The needed LPCs in (3.3.19) can be computed in numerous ways. We will in the following use the Burg algorithm, as it will produce spectral estimates with higher resolution than, for instance, the solution to the Yule-Walker equations would [UB75]. To mitigate the slight phase-dependent frequency bias introduced by the Burg algorithm, we make use of the weighting technique presented in [Swi80] in our simulations (see also [Mar87] for a further discussion on this topic). The resulting estimator is denoted the BASC spectral estimator.

Note that the BASC estimator will produce (almost) the same estimate as the ASC estimator computed from the forward-backward averaged covariance matrix estimate in (3.2.9). This forward-backward ASC (FB-ASC)
will yield a significantly better spectral estimate, although with a somewhat lower resolution, than the forward-only ASC (F-ASC) estimator (see also [LLS98, JS99]). The reason that BASC will produce only almost the same estimate as FB-ASC can be explained as follows: the estimate of \( C \), as obtained by using either the Burg or the Levinson-Durbin algorithm, will in general not yield the same estimate of \( C \) as the Cholesky factorization of the covariance matrix estimate in (3.2.9). This is due to the fact that \( \hat{R} \) is the outer product covariance matrix estimate, and will thus not have the Toeplitz structure that is obtained if an estimate of \( \hat{R} \) is constructed from a \( C \) computed by the Burg or the Levinson-Durbin algorithm. Thus, the BASC and the FB-ASC spectral estimates will not be identical, although as is shown in the next section the estimates are basically the same.

### 3.4 Numerical examples

As the finite-sample performance analysis of the spectral estimators is quite difficult at best, we illustrate the estimators’ performance using numerical simulations. The true spectrum of the simulation data, as is shown in Figure 3.1(a), consists of three dominant spectral lines and ten smaller spectral lines located at the following frequencies: 0.0625, 0.0725, 0.25, 0.28, 0.33, 0.35, 0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49. These spectral lines all have a phase offset of \( \pi/4 \). The data sequence has \( N = 64 \) data samples and is corrupted by complex white Gaussian noise with zero mean and variance \( \sigma^2 \). Figure 3.1 and Figure 3.2 illustrate the resolution and the accuracy of the Periodogram, the forward-only PSC (F-PSC), the F-ASC, the FB-ASC and the BASC estimators. As seen from the figure, the Periodogram and the F-PSC estimator, as shown in Figure 3.1(b)-(c), will both fail to resolve the first two closely spaced peaks. Similarly, the FB-PSC as well as the PSC implementation proposed by Musicus, will also fail to resolve the peaks. The F-ASC, the FB-ASC (both implemented using the technique presented in Liu et al [LLL98]) and the BASC estimators, as shown in Figure 3.1(d) and Figure 3.2(a)-(b), will all resolve the peaks. Note that the FB-ASC and the BASC, as shown in Figure 3.2(a)-(b), will yield almost the same spectral estimates, even though they are based on two different covariance matrices. The main difference being that the BASC estimate is seen to have somewhat more accurate amplitude estimates. In the example the SNR = 15 dB, and the filter was 24 taps long.
Figure 3.1: Illustration of the resolution and accuracy for the different spectral estimators. (a) The true spectrum. (b) The Periodogram. (c) The F-PSC spectrum. (d) The F-ASC spectrum.
Figure 3.2: Illustration of the resolution and accuracy for the different spectral estimators. (a) The FB-ASC spectrum. (b) The BASC spectrum.

We proceed by studying the different spectral estimators’ resolution in more detail by varying the location of the second spectral line. Let $P(\omega)$ denote an amplitude spectral estimator at frequency $\omega$. It is common practice in spectral analysis to quantify a method’s capability to resolve two sinusoidal signals using the rule (see, e.g., [KB86, Zha98])

$$\gamma = 2P(\omega_3) - P(\omega_1) - P(\omega_2) < 0,$$

where $\omega_3$ is the midway frequency between $\omega_1$ and $\omega_2$, i.e.,

$$\omega_3 = \frac{1}{2}(\omega_1 + \omega_2).$$

The two signals are said to be resolvable if $\gamma$ is negative and irresolvable otherwise. Figure 3.3 shows the resolution ability of the F-ASC, the BASC (the FB-ASC has the same resolution), the PSC and the FB-PSC spectral estimators as compared to the standard Periodogram as well as the PSC implementation proposed by Musicus [Mus85]. The resolution of the parametric spectral MUSIC algorithm, assuming full knowledge of the data structure, is also shown as a comparison. Note that MUSIC requires knowledge of the number of sinusoidal components. The filterbank methods do not assume this knowledge. Still, it is worth noting that the MUSIC method does not have a higher resolution than the F-ASC or the BASC methods.
Figure 3.3: Resolution limits.

As seen from the figure, the BASC (and the FB-ASC) will have somewhat lower resolution than the F-ASC, both having clearly higher resolution than the other methods. Also worth noting is that the FB-PSC has, in contrast to the F-ASC and the FB-ASC, higher resolution than the F-PSC, and that the PSC implementation proposed by Musicus has a severe resolution problem. In the example the SNR=20 dB, and the filter was 24 taps long. Our results were obtained from 500 Monte Carlo trials.

Figure 3.4: Relative computational complexity vs data length.

Next, we study the computational complexity of the different methods, as the data length, \(N\), varies. Figure 3.4 illustrates the relative computational complexity of the F-PSC, the FB-PSC, F-ASC, the FB-ASC (all obtained by using the implementations in [LLL98]) and the BASC algorithms. The computational complexity of the MUSIC algorithm is also shown as a comparison. In the figure, the estimators' computational load, as measured by MATLAB, has been normalized with the load of the FB-ASC method.
As seen from the figure, the BASC estimator will, for $N = 256$, be about twice as fast as the FB-ASC implementation presented in [LLL98]. We remind the reader that the only difference between the two methods is how the estimate $\hat{C}$ is computed. For $N = 256$, the direct calculation of the forward-backward covariance matrix, its inverse and the Cholesky factorization of the latter accounts for roughly 55% of the total number of flops as compared to 4% in our approach. In the example, the filterlength was $L = N/4$, and the spectrum was evaluated for $4N$ grid points.

### 3.5 Conclusions

In this chapter, we have presented an efficient implementation of the Amplitude Spectrum Capon (ASC) estimator, denoted the BASC estimator. The implementation is based on the Burg algorithm, and can be efficiently computed using the FFT. The BASC algorithm is found to be significantly faster than the recent algorithm by Liu et al, especially for longer data sequences.
Chapter 4

Combining Capon and APES for Estimation of Spectral Lines

We propose a combining of the Capon- and the APES spectral estimators for estimation of both the amplitude and the frequency of spectral lines. The so-obtained estimator does not suffer from Capon’s biased amplitude estimates nor from APES’ biased frequency estimates or resolution problem. Furthermore, the combined estimator is computationally simpler than APES, and has about the same complexity as Capon. Numerical simulations are presented illustrating the increased performance.

4.1 Introduction

There has of lately been an increased interest in non-parametric spectral estimation using the recently proposed APES (Amplitude and Phase ESTimation) spectral estimator. The APES method, which is a Capon-like method using an estimate of the noise covariance matrix, was originally derived by means of a relatively involved approximate maximum likelihood approach [LS96a]. In Chapter 2, it was found that the APES method, as well as the well-known Capon method [Cap69, Lac71], could be interpreted as members of the class of matched-filterbank (MAFI) spectral estimators. With the MAFI interpretation, the APES estimator was given a simpler and more intuitive derivation, which was later extended to a form using forward-backward averaging [LSLJ97, LLS98], as well as to the case of real-valued data (see Chapter 5). Furthermore, there has been recent work wherein efficient implementations of the Capon and the APES estimators
was proposed [LLL98] (see also Chapter 3), as well as a new derivation of the APES estimator from pure narrowband filter design considerations [SLL99].

It was found in [LS96a] that the APES estimator in general gives a significantly more accurate amplitude and phase estimate of the spectrum than the Capon estimator does. In Chapter 2, this was further examined and it was shown that the Capon estimator, to within a second-order approximation, will be biased (downward) whereas the APES estimator will be unbiased. For data containing spectral peaks this means that at the true peak locations the APES estimator will outperform the Capon estimator. As the peak locations are normally assumed to be unknown, the spectrum is estimated over a grid of frequencies and the sinusoidal components are found as the spectral peaks.

In general, the peak locations in the APES spectral estimate will not be located at the true frequencies. In fact, in this chapter we show that the APES estimates of the peak locations are in many cases biased, and that the corresponding peak amplitude estimate in these cases will also be (slightly) biased. Furthermore, it is shown that the APES estimates will have lower resolution than the Capon estimates. To circumvent these disadvantages, we propose an alternative estimation procedure first estimating the peak locations using the Capon estimator, and then refining the amplitude and phase estimates at these frequencies using the APES estimator. The resulting estimator, termed CAPES, is found to give significantly more accurate amplitude and phase estimates than both the Capon and the APES estimators. As the CAPES estimator is a non-parametric approach, it does not require detailed knowledge of how many sinusoidal components the data contains. This is in stark contrast to parametric methods, such as MUSIC [Sch81] (see also [Hän99]) or ESPRIT [RK89].

The chapter is organized as follows: In the next section the MAFI interpretation of the Capon and APES spectral estimators, as was presented in Chapter 2, is briefly reviewed. In Section 4.3, a natural formulation of a frequency estimator based on weighted least-squares is presented, in Section 4.4 the CAPES estimator is introduced and an efficient implementation is proposed. In Section 4.5, we illustrate and compare the performance of the different estimators. Section 4.6 contains our concluding remarks.
4.2 The Matched Filterbank Approach

In the filterbank approach to spectral estimation, the spectrum is estimated by passing the signal through a narrowband filter, $h_\omega$, with varying center frequency $\omega$ (see, e.g., [LS96a, SM97]). Here, and in the following, the subscript $\omega$ is used to indicate a dependence on the filter’s center frequency.

Let $\{x(t); t = 1, \ldots, N\}$ denote the available (stationary) data sample of which the spectrum is to be estimated. The filter output can then be written as

$$h_\omega^* y(t) = \alpha_\omega e^{i\omega t} + \epsilon(t), \quad t = 1, \ldots, M$$  \hspace{1cm} (4.2.1)

where $(\cdot)^T$ and $(\cdot)^*$ denote transpose and complex conjugate transpose, $M = N - L + 1$,

$$y(t) = [ x(t) \quad \ldots \quad x(t + L - 1) ]^T,$$  \hspace{1cm} (4.2.2)

and where $\epsilon(t)$ is some additive colored noise. In the following we assume that $\epsilon(t)$ is uncorrelated with $\alpha_\omega e^{i\omega t}$. This assumption is basically only true for spectral lines, although it has been found to be a reasonable approximation in other cases as well [LS96a].

The least-squares estimate of the complex amplitude, $\alpha_\omega$, in (4.2.1) is given by

$$\hat{\alpha}_\omega = h_\omega^* Y_\omega,$$  \hspace{1cm} (4.2.3)

where

$$Y_\omega = \frac{1}{M} \sum_{t=1}^{M} y(t) e^{-i\omega t}.$$  \hspace{1cm} (4.2.4)

The choice of filterlength, $L$, should be done by a compromise between resolution and statistical stability: the larger $L$ the better the resolution but the worse the statistical stability. It is not difficult to understand why this is so. The time-bandwidth product type of result (see, e.g., [SM97]) implies that the filter can be made narrower in the frequency domain, and hence the corresponding spectral resolution enhanced, as $L$ increases. On the other hand, $M$ decreases with increasing $L$ and therefore we have fewer samples for the estimation of $\alpha_\omega$ in (4.2.3) and (4.2.4), which should lead to a poorer statistical accuracy.

The problem of designing the filter $h_\omega$ as a matched-filterbank (MAFI) was studied in Chapter 2. It was found that both the Capon method and the APES method can be interpreted as members of the MAFI class. The corresponding filters are given by

$$h_\omega^{\text{Capon}} = R^{-1} a_\omega \left( \alpha_\omega^* \hat{R}^{-1} a_\omega \right)^{-1}$$  \hspace{1cm} (4.2.5)

$$h_\omega^{\text{APES}} = Q^{-1} a_\omega \left( \alpha_\omega^* \hat{Q}^{-1} a_\omega \right)^{-1},$$  \hspace{1cm} (4.2.6)
where \( \hat{Q}_\omega \) is an estimate of the noise covariance matrix, found as (see Chapter 2 for more details)

\[
\hat{Q}_\omega = \hat{R} - Y_\omega Y_\omega^*, \tag{4.2.7}
\]

and

\[
\hat{R} = \frac{1}{M} \sum_{t=1}^{M} y(t)y(t)^*, \tag{4.2.8}
\]

\[
a_\omega = \begin{bmatrix}
e^{i\omega} & \cdots & e^{i(L-1)\omega}
\end{bmatrix}^T. \tag{4.2.9}
\]

The Capon and APES spectral amplitude estimates, at frequency \( \omega \), are given by (4.2.3) evaluated using the filter (4.2.5) and (4.2.6), respectively. The so-obtained APES spectral estimate will have lower resolution than the Capon spectral estimate, but will give a significantly better amplitude and phase estimate at the locations of the true spectral peaks (see Chapter 2). The \( k \)th spectral peak frequency estimate, \( \hat{\omega}_k \), is found as

\[
\hat{\omega}_k = \arg \max_k |h_\omega^* Y_\omega|, \tag{4.2.10}
\]

where \( \max_k \) refers to the \( k \)th maximum. However, as is illustrated in Section 4.5, the locations of the APES spectral peaks will be somewhat biased, and the performance of the corresponding peak amplitude and phase estimates will be worse than at the true peak locations.

The problem is thus to first estimate the peak locations, and then refine the amplitude and phase estimates for these locations.

### 4.3 Weighted Least-Squares

A natural formulation of the spectral peak estimation problem is given by the weighted least-squares (WLS) approach

\[
\min_{\omega, \alpha} \sum_{t=1}^{M} \| y(t) - \alpha_\omega a_\omega e^{i\omega t} \|^2 / \hat{Q}_\omega^{-1}. \tag{4.3.11}
\]

A simple calculation gives

\[
\hat{\alpha}_\omega^{WLS} = \hat{\alpha}_\omega^{APES}, \tag{4.3.12}
\]

which after insertion in (4.3.11), yields

\[
\hat{\omega}_k^{WLS} = \arg \max_k \left\{ a_\omega^* \hat{Q}_\omega^{-1} a_\omega |\hat{\alpha}_\omega^{APES}|^2 \right\}. \tag{4.3.13}
\]
Thus, interestingly enough, the WLS amplitude estimate will coincide with the APES amplitude estimate, but the frequency estimates will differ. This fact in itself gives an indication that the APES frequency estimates are most likely not "optimal". It is our experience that the presented WLS frequency estimator outperforms the APES frequency estimator. Another alternative is to use the Capon frequency estimator obtained by combining (4.2.10) and (4.2.5). We have found that this will give even better frequency estimates than WLS.

In the next section we thus suggest a two-step estimator combining the Capon frequency estimates with the APES amplitude estimates. The so-obtained estimator, termed CAPES, is found to give significantly more accurate amplitude and phase estimates than both the Capon and the APES estimators.

4.4 CAPES

The CAPES algorithm first estimates the spectral peak locations using the (forward-only)\(^1\) Amplitude Spectrum Capon (F-ASC) spectrum estimator given by (4.2.10) and (4.2.5), i.e.,

\[
\hat{\omega}_k^{\text{Capon}} = \arg \max_k \left| \hat{\omega}_k^{\text{Capon}} \right|
\]

\[
= \arg \max_k \left| \frac{\lambda_\omega}{\psi_\omega} \right|
\]

(4.4.14)

where

\[
\lambda_\omega = a_\omega \hat{R}^{-1} y_\omega
\]

(4.4.15)

\[
\psi_\omega = a_\omega^* \hat{R}^{-1} a_\omega.
\]

(4.4.16)

It is worth stressing that the frequency estimator in (4.4.14) will yield better frequency estimates than the peak locations of the "standard" Power Spectrum Capon (PSC) spectral estimator \(h^*_\omega \hat{R} h_\omega\) [Cap69, Lac71] (see also Chapter 3). The reason for this improvement is that the estimator in (4.4.14) is the least-squares estimate of \(\alpha_\omega\) in (4.2.1), whereas the PSC estimator will estimate \(|\alpha_\omega|^2\) ignoring the additive noise.

As a second step, the CAPES algorithm refines the spectral amplitude and phase estimates for the estimated spectral peak frequencies, \(\hat{\omega}_k^{\text{Capon}}\),

\(^1\) It is our experience that the forward-only ASC estimator has somewhat better spectral resolution than the forward-backward averaged Amplitude Spectrum Capon (FB-ASC) estimator (see also Figure 3.3).
using the APES algorithm. The forward APES (F-APES) estimate, obtained by combining (4.2.3) and (4.2.6), can be found as

\[
\hat{\alpha}_{\omega_k}^{F-APES} = \frac{a_{\omega}^* \left( \hat{R} - Y_\omega Y_\omega^* \right)^{-1} Y_\omega}{a_{\omega}^* \left( \hat{R} - Y_\omega Y_\omega^* \right)^{-1} a_{\omega}} = \frac{a_{\omega}^* \left( \hat{R} - Y_\omega Y_\omega^* \right)^{-1} a_{\omega}}{a_{\omega}^* \left( \hat{R} - Y_\omega Y_\omega^* \right)^{-1} Y_\omega + a_{\omega}^* \hat{R}^{-1} Y_\omega}
\]

where

\[
\gamma_\omega = Y_\omega^* \hat{R}^{-1} Y_\omega.
\]

It is worth noting that the forward-backward averaged APES (FB-APES) estimator will yield somewhat better amplitude estimates than the presented forward-only estimator, although to a higher computational cost (see [LSLJ97, LLS98] for details on the derivation and the implementation of the forward-backward estimator).

### 4.5 Numerical Examples

In this section we will numerically study the accuracy and resolution of the ASC, the APES and the CAPES spectral estimators in a number of cases.

The true spectrum of the simulated sinusoidal signal is shown in Figure 4.1(a). It consists of three dominant spectral lines and ten smaller spectral lines located at the following frequencies: 0.0625, 0.0725, 0.25, 0.28, 0.33, 0.35, 0.37, 0.39, 0.41, 0.43, 0.45, 0.47, 0.49. These spectral lines all have a phase offset of \(\pi/4\). The data sequence has \(N = 64\) data samples and is corrupted by complex white Gaussian noise with zero mean and variance \(\sigma^2\). The SNR for the \(k\)th spectral line is defined as

\[
\text{SNR}_k = 10 \log_{10} \left( \frac{|\alpha_k|^2}{\sigma^2} \right) \text{ [dB]},
\]

where \(\alpha_k\) is the complex amplitude of the \(k\)th spectral line.
Figure 4.1: Illustration of the resolution and accuracy for the different spectral estimators. Note the different scales for WLS and F-APES. (a) The true spectrum. (b) The Periodogram. (c) The WLS “pseudospectrum” for estimating the peak frequencies as given in (4.3.13). (d) The F-APES spectrum.
Figure 4.2: Illustration of the resolution and accuracy for the different spectral estimators. (a) The F-ASC spectrum. (b) The CAPES spectrum of the first 4 peaks.

Figure 4.1 and Figure 4.2 illustrates the resolution and the accuracy of the proposed estimator for a randomly chosen data realization. As seen in Figures 4.1(b)–(d), the Periodogram, the WLS frequency estimator presented in (4.3.13), and the F-APES spectral estimator will all fail to clearly resolve the two first closely spaced peaks. The F-ASC’s superior resolution is shown in Figure 4.2(a), but the spectral amplitude estimates can be seen to be rather poor. The amplitude estimates for these spectral lines will be significantly improved by the CAPES spectral estimator as shown in Figure 4.2(b). In the example the SNR\(_1\) = 10 dB, and the filter was 24 taps long.

We proceed by studying the different spectral estimators’ resolution in more detail by varying the location of the second spectral line. Let \(P(\omega)\) denote an amplitude spectral estimator at frequency \(\omega\). It is common practice in spectral analysis to quantify a method’s capability to resolve two sinusoidal signals using the rule (see, e.g., [KB86, Zha98])

\[
\gamma = 2P(\omega_3) - P(\omega_1) - P(\omega_2) < 0,
\]

where \(\omega_3\) is the midway frequency between \(\omega_1\) and \(\omega_2\), i.e.,

\[
\omega_3 = \frac{1}{2}(\omega_1 + \omega_2).
\]

The two signals are said to be resolvable if \(\gamma\) is negative and irresolvable oth-
Figure 4.3: Resolution limits.

erwise. Figure 4.3 shows the resolution ability of the F-ASC (the FB-ASC will have slightly lower resolution, see Figure 3.3), F-APES, FB-APES, and the WLS spectral estimator as compared to the standard Periodogram as well as the parametric spectral MUSIC algorithm which assumes full knowledge of the data structure. Note that MUSIC requires knowledge of the number of sinusoidal components. The filterbank methods do not assume this knowledge. Still, it is worth noting that the MUSIC method does not have a higher resolution than the F-ASC method. As seen from the figure, the F-ASC estimator has the best resolution, followed closely by the MUSIC and the WLS estimators. The FB-APES estimator is found to have slightly higher resolution than the Periodogram, whereas the F-APES has a lower resolution than the Periodogram\(^2\). In the example the SNR\(_1\) = 20 dB, and the filter was 24 taps long. Our results were obtained from 500 Monte Carlo trials.

Next, we study the computational complexity of the different methods, as the data length, \(N\), varies. Figure 4.4 illustrates the computational complexity of MUSIC, the F-APES and the F-ASC algorithms. The latter two are computed using the algorithms presented in [LLL98]. The figure shows the number of floating point operations (flops) as measured by MATLAB for the calculation of the different spectral estimates. The CAPES algorithm, needing also the computation of the refined F-APES amplitude and phase estimates for the peak locations in the F-ASC spectrum, requires about the same amount of computations as the F-ASC algorithm (the dif-
ference is negligible). In the example, the filterlength was \( L = N/4 \), and the spectrum was evaluated for \( 4N \) grid points.

![Graph](image)

**Figure 4.4:** Computational complexity vs data length.

Finally, we compare the mean squared error (MSE) and the bias of the different frequency and amplitude estimators. To make the comparison possible, we study an example when all the spectral lines are resolved by all the estimators. This is done by choosing the second spectral line frequency as 0.085 instead of 0.0725 as in the examples above. Figure 4.5(a) compares the MSE of the frequency estimates of the first spectral line (the others behave similarly) for the WLS, the F-ASC, the F-APES, the spectral MUSIC frequency estimators as well as the Periodogram as the SNR varies. As seen from the figure, the F-ASC frequency estimator will have the best performance after the Periodogram (the reader is reminded that F-ASC has a much better resolution than the Periodogram). Note also that the Periodogram will be an efficient estimator achieving the CRB when the peaks are well resolved. Similarly, Figure 4.5(b) compares the bias of the frequency estimates for the different estimators. In Figures 4.5(c) and (d) the MSE and the bias of the amplitude estimates for the F-APES, the F-ASC, and the CAPES spectral estimators as well as the Periodogram are shown as the SNR varies. From these figures it is clear that the CAPES estimator has lower MSE and less bias than any of the other spectral estimators.
Figure 4.5: The figure shows the (a) the frequency MSE (b) the frequency bias (c) the amplitude MSE (d) the amplitude bias as the SNR varies.
In Figures 4.6(a) and (b) the estimators MSE and bias are similarly shown as the filterlength, \( L \), varies. Remember that the resolution for the filterbank-based spectral estimators will improve as the filterlength increases. Notice that the choice of filterlength is significantly simplified for the CAPES spectral estimator that has similar performance for most filterlengths. Also note that the CAPES estimator can achieve a higher resolution (i.e., longer filterlengths) before losing accuracy. In this example \( \text{SNR}_1 = 20 \text{ dB} \) and the results were obtained from 500 Monte Carlo trials.

### 4.6 Conclusions

In this chapter, we have presented a combined Capon and APES spectral estimator for estimation of both amplitude and frequency of spectral lines. The combined estimator, termed the CAPES estimator, uses the Amplitude Spectrum Capon (ASC) spectral estimator’s superior frequency estimation to locate the frequencies of the spectral lines, and then refines the amplitude and phase estimates for these frequencies using the APES spectral estimator. The resulting estimator has about the same computational complexity as the Capon estimator, and is computationally simpler than the APES estimator.
Chapter 5

Capon and APES Spectrum Estimation for Real-Valued Signals

This chapter considers the problem of estimating the spectrum of real-valued signals. We propose real-valued versions of the Capon and the APES spectral estimators. The estimators are derived as members of the Matched-Filterbank (MAFI) estimator class as introduced in Chapter 2. Furthermore, we show that the real-valued estimators will be unbiased, whereas the complex-valued estimators will have a (slight) bias for real-valued data. Finally, we conclude the chapter with a numerical example illustrating the performance of the proposed estimators.

5.1 Introduction

In the filterbank approach to spectral estimation, the amplitude of the spectrum is estimated by passing the signal through a narrowband filter, \( h_\omega \), with varying center frequency \( \omega \) (see, e.g., [LS96a, LSGM86]). Here, and in the following, the subscript \( \omega \) is used to indicate a parameter's dependence on the filter's center frequency.

Let \( \{x(t); t = 1, \ldots, N\} \) denote the available (stationary) data sample of which the spectrum is to be estimated, where \( N \) denotes the number of data samples. The filter output can then be written as

\[
h_\omega y(t) = a(\omega)e^{j\omega t} + \epsilon(t), \quad t = 1, \ldots, M
\]  

(5.1.1)

where \((\cdot)^T\) and \((\cdot)^*\) denote transpose and complex conjugate transpose,
\[ M = N - L + 1, \]
\[
y(t) = [x(t) \quad \ldots \quad x(t + L - 1)]^T,
\]
and \( \epsilon(t) \) is some additive colored noise. The least-squares estimate of the complex amplitude, \( \hat{a}(\omega) \), in (5.1.1) is then given by
\[
\hat{a}(\omega) = h^*_\omega Y_\omega
\]
where
\[
Y_\omega = \frac{1}{M} \sum_{t=1}^{M} y(t)e^{-j\omega t}.
\]

The problem of designing \( h_\omega \) as a matched-filterbank (MAFI) was studied in Chapter 2. It was found that the well-known Capon method [Cap69, Lac71], as well as the recently proposed APES (Amplitude and Phase Estimation) method [LS96a], can be interpreted as members of the MAFI class. The corresponding filters are designed as
\[
h_{\omega}^{\text{Capon}} = \frac{\hat{R}^{-1}a_\omega}{a^*_\omega \hat{R}^{-1}a_\omega}
\]
\[
h_{\omega}^{\text{APES}} = \frac{\hat{Q}_\omega^{-1}a_\omega}{a^*_\omega \hat{Q}_\omega^{-1}a_\omega}
\]
where the estimate of the (complex-valued data) noise covariance matrix, \( \hat{Q}_\omega^c \), is found as (see Chapter 2)
\[
\hat{Q}_\omega^c = \hat{R} - Y_\omega Y_\omega^*.
\]
and
\[
\hat{R} = \frac{1}{M} \sum_{t=1}^{M} y(t)y(t)^*
\]
\[
a_\omega = \begin{bmatrix} 1 & e^{i\omega} & \ldots & e^{i(L-1)\omega} \end{bmatrix}^T.
\]

Many signals are actually real-valued which the above filter design does not exploit. For such signals, it is more appropriate to design the filter taking into account the fact that the spectrum is symmetric.

In the following section we present a real-valued version of the Capon and APES spectral estimators. In Section 5.3, we discuss these estimators bias in the case of real-valued data. Section 5.4 contains our numerical simulations illustrating the proposed estimators performance, and Section 5.5 contains our conclusions.
5.2 The Real-Valued MAFI Approach

It is reasonable to expect better performance of the real-valued filter design because by construction the filters will pass both frequencies of interest, $\omega$ and $-\omega$, undistorted, whereas the complex-valued design above will only pass $\omega$ undistorted, and will try to null $-\omega$, therefore yielding less power in the filtered output.

The filterbank approach basically reduces the problem of estimating the spectrum of $x(t)$ to that of estimating the amplitude of a sinusoidal signal buried in colored noise (see, e.g., [LS96a, LSGM86]). Therefore, under the assumption of real-valued data, it is reasonable to consider $x(t)$ to be additively decomposed as

$$x(t) = r_\omega \sin(\omega t + \varphi_\omega) + n(t), \quad (5.2.10)$$

where $r_\omega \in \mathbb{R}$, $\omega \in (0, 2\pi]$ and $n(t)$ is some additive colored noise. By rewriting (5.2.10) as

$$x(t) = a_\omega \cos(\omega t) + b_\omega \sin(\omega t) + n(t), \quad (5.2.11)$$

where

$$\varphi_\omega = \arctan \left( \frac{a_\omega}{b_\omega} \right) \quad (5.2.12)$$

$$r_\omega = \sqrt{a_\omega^2 + b_\omega^2}, \quad (5.2.13)$$

we can write $y(t)$, as defined in (5.1.2), as

$$y(t) = A_\omega u(t) + n(t), \quad (5.2.14)$$

where $n(t)$ is formed in the same way as $y(t)$, and

$$A_\omega = \begin{bmatrix}
1 & 0 \\
\cos \omega & \sin \omega \\
\vdots & \vdots \\
\cos(L-1)\omega & \sin(L-1)\omega
\end{bmatrix} \quad (5.2.15)$$

$$u(t) = \begin{bmatrix}
\cos(\omega t) & \sin(\omega t) \\
-\sin(\omega t) & \cos(\omega t)
\end{bmatrix} \theta_\omega \quad (5.2.16)$$

\[ \Delta \Theta_\omega v_T^T, \quad (5.2.17) \]

where

$$\theta_\omega = \begin{bmatrix} a_\omega & b_\omega \end{bmatrix}^T \quad (5.2.18)$$

$$\Theta_\omega = \begin{bmatrix} a_\omega & b_\omega \\
b_\omega & -a_\omega \end{bmatrix} \quad (5.2.19)$$

$$v_t = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \end{bmatrix}. \quad (5.2.20)$$
As in the complex-valued case (see Chapter 2), the choice of the filter length, \( L \), should be done by a compromise between resolution and statistical stability: the larger \( L \) the better the resolution but the worse the statistical stability.

Following the MAFI design in Chapter 2, the real-valued MAFI filter is designed such that the corresponding signal-to-noise ratio (SNR) in the filter’s output is maximized, i.e.,

\[
\max_{h_\omega} \frac{\| h_\omega^T A_\omega \|^2}{h_\omega^T Q_\omega^{-1} h_\omega} \tag{5.2.21}
\]

where \( \| \cdot \| \) denotes the vector norm. The filter is constrained as (cf. (5.2.14) and (5.2.16))

\[
h_\omega^T A_\omega = \begin{bmatrix} 1 & 0 \end{bmatrix} \triangleq c^T, \tag{5.2.22}
\]

and \( \hat{Q}_\omega \) is an estimate of the (real-valued data) noise covariance matrix. The maximization in (5.2.21), under the constraint (5.2.22), can easily be found to be equivalent to

\[
\min_{h_\omega} h_\omega^T \hat{Q}_\omega^{-1} h_\omega \quad \text{subject to} \quad h_\omega^T A_\omega = c^T, \tag{5.2.23}
\]

which is a well-known, well-studied, minimization problem (see, e.g., [SM97]). The real-valued version of the Capon and APES estimators (termed rv-Capon and rv-APES) can then be found as (cf. (5.1.5) and (5.1.6))

\[
h_{\omega,rv-Capon} = \hat{R}^{-1} A_\omega (A_\omega^T \hat{R}^{-1} A_\omega)^{-1} c \tag{5.2.24}
\]

\[
h_{\omega,rv-APES} = \hat{Q}_\omega^{-1} A_\omega (A_\omega^T \hat{Q}_\omega^{-1} A_\omega)^{-1} c. \tag{5.2.25}
\]

Using (5.2.17), the estimate of the data covariance matrix can be written as

\[
\hat{R} = (\widehat{A}_\omega \widehat{\Theta}_\omega) G (\widehat{A}_\omega \widehat{\Theta}_\omega)^T + \hat{Q}_\omega \tag{5.2.26}
\]

where

\[
G = \frac{1}{M} \sum_{t=1}^M v_t^T v_t. \tag{5.2.27}
\]

From (5.2.14), the least squares estimate of \( A_\omega \Theta_\omega \), ignoring the fact that \( A_\omega \) is known, is given by

\[
(\widehat{A}_\omega \widehat{\Theta}_\omega) = \hat{C}_\omega G^{-1} \tag{5.2.28}
\]
where
\[ \hat{C}_\omega = \frac{1}{M} \sum_{i=1}^{M} y(t)v_i. \] (5.2.29)

Finally, inserting (5.2.28) into (5.2.26) yields the following estimate of \( \hat{Q}_\omega \) (cf. (5.1.7))
\[ \hat{Q}_\omega^r = \hat{R} - \hat{C}_\omega G^{-1}\hat{C}_\omega^T. \] (5.2.30)

By using equation (5.2.22) and the structure in (5.2.16), a least-squares estimator of \( \theta_\omega \) can then be formulated as
\[ \hat{\theta}_\omega = G^{-1}\hat{C}_\omega^T h_\omega, \] (5.2.31)
where \( h_\omega \) denotes either the Capon or the APES-filter in (5.2.24) or (5.2.25), respectively. Then take \( r_\omega \), obtained by combining (5.2.13) and the estimates in (5.2.31), as an estimate of the spectrum of \( x(t) \) at frequency \( \omega \).

From a computational viewpoint, it is here worth mentioning that the calculation of \( Q_\omega^{-1} \) in (5.2.25) can be done efficiently by expanding the inversion of \( Q_\omega^r \) using the matrix inversion lemma (see, e.g., [SM97]). Furthermore, note that \( G^{-1} \) in (5.2.30) can be calculated in a closed form expression and that the sum in (5.2.29) can be efficiently computed using the real and imaginary part of the FFT.

Also, note that in the cases where \( \sin(\omega t) \approx 0 \) (and similarly for \( \cos(\omega t) \)) the inversions needed to calculate (5.2.24), (5.2.25) and (5.2.31) will need special solutions (which are easily derived) due to the fact that the matrices will be singular.

### 5.3 Bias Analysis

To examine the behavior of the complex-valued Capon and APES spectral estimators on real-valued data, we reformulate the data model in (5.2.10) using Euler’s formulas. The sample vector \( y(t) \) in (5.1.2) can then be expressed as
\[ y(t) = \alpha_\omega a_\omega e^{i\omega t} + \alpha_{-\omega}^* a_{-\omega} e^{-i\omega t} + n(t), \] (5.3.32)
where \( \alpha_\omega = (r_\omega/2) e^{i(\varphi_\omega - \pi/2)} \). The least squares estimator in (5.1.3) can then be written as
\[ \hat{\alpha}_\omega = \alpha_\omega + \frac{L}{M} \alpha_{-\omega}^* e^{i\omega} H_{\omega}(-\omega) \rho(M, \omega) + h_\omega^* \Delta_\omega, \] (5.3.33)
where \( (\cdot) \) denotes complex conjugate. The frequency response of the filter \( h_\omega \) at frequency \( f \), \( H_\omega(f) \), is found as
\[ H_\omega(f) = \frac{1}{L} \sum_{k=1}^{L} h_\omega(k)e^{-ifk} = \frac{1}{L} e^{-if} \hat{h}_\omega^* \pi_f, \] (5.3.34)
and, where
\[
\rho(M, \omega) = \sum_{t=1}^{M} e^{i2\omega t}
\]
\[
= e^{i(M+1)\omega} \frac{\sin(M\omega)}{\sin(\omega)}
\]
\[
\Delta_{\omega} = \frac{1}{M} \sum_{t=1}^{M} n(t)e^{-i\omega t}.
\]

Following the discussion in Chapter 2, the expected value for the estimation error in (5.33) can be expressed as
\[
E(\hat{\theta}_\omega - \theta_\omega) = \frac{L}{M} \alpha_\omega^* e^{i\omega} H_\omega(-\omega) \rho(M, \omega),
\]

where \(E(\cdot)\) denotes expectation. Since \(\rho(M, \omega)\) in (5.38) is an oscillating function, and as \(H_\omega(-\omega)\), although small, will be non-zero, the complex-valued estimators will have a (slight) bias (at most frequencies) for real-valued data. It should be noted that, even though the methods are biased on real-valued signals, this bias will in most cases be small. The methods will treat the contribution from \(\alpha_\omega^* e^{-i\omega t}\) as part of the noise and will thus try to minimize its influence.

A similar procedure for the least-squares estimate for the real-valued spectral estimators will indicate that they are unbiased. The estimation error in (5.2.31) can, by inserting the expression for \(C_\omega\) in (5.2.29), be rewritten as
\[
E \left( \theta_\omega - \theta_\omega \right) = E \left( G^{-1} \frac{1}{M} \sum_{t=1}^{M} v_t^T v_t \theta_\omega \right) +
\]
\[
+ E \left( G^{-1} \frac{1}{M} \sum_{t=1}^{M} v_t^T h_\omega^T n(t) \right) - \theta_\omega
\]
\[
= \theta_\omega + 0 - \theta_\omega = 0
\]

The estimate of the spectrum, \(\hat{r}_\omega\), as given by (5.2.13), will thus be unbiased.
5.4 Numerical Examples

As the finite-sample performance analysis of the spectral estimators is quite difficult at best, we illustrate the estimators’ performance using numerical simulations. In this section we study how the estimators performance depends on the signal-to-noise ratio (SNR). Figure 5.1(a)–(b) shows the modulus of the true spectrum which consists of three superimposed real-valued sinusoids located at the following frequencies: 0.1475, 0.1850, 0.2194 (where 1 is the sampling frequency).

![Figure 5.1: (a) The modulus of the true spectrum, plotted together with (a) the Periodogram and (b) the rv-APES spectral estimate, SNR=25dB.](image)

The sinusoids have initial phases $0.2\pi, 0.6\pi, 0.2\pi$. The data sequence has $N = 64$ data samples and is corrupted by a zero mean white Gaussian noise with standard deviation $\sigma$. In the simulated data these were 0.3, 2, 1. The filterlength, $L$, was set to 20 taps. To obtain a reasonably fine grid on the frequency axis the spectrum was evaluated at 256 grid points. In Figure 5.1(a) the limited amount of data gives a corrupted Periodogram, while the rv-APES in Figure 5.1(b) manage to reproduce the true spectrum quite accurately.

Figures 5.2(a) and (b) shows the bias and the mean square error (MSE) of the Capon (solid line), rv-Capon (dashed line), APES (dash-dotted line) and rv-APES (dotted-line) spectral estimators for the second frequency (the others behave similarly) as the SNR varies. As was expected the real-valued
estimators have a somewhat lower bias and MSE than their complex-valued counterparts. Note that the reason why the MSE increases for high SNR is that \( \mathbf{R} \) becomes almost singular.

It should be mentioned that the estimators performance varies somewhat with the frequency separation, but the case shown is typical. Our results are obtained from 1000 Monte Carlo trials.

5.5 Conclusions

In this chapter, we consider the problem of estimating the spectrum of real-valued signals. We propose real-valued versions of the Capon and the APES spectral estimators, and show that the real-valued estimators will be unbiased, whereas the complex-valued estimators will have a (slight) bias for real-valued data. Numerical examples illustrate the performance of the proposed estimators.
Chapter 6

Two-Dimensional Capon Spectral Analysis

We present a computationally efficient algorithm for computing the 2-D Capon spectral estimator. The implementation is based on the fact that the 2-D data covariance matrix will have a \textit{Toeplitz-Block-Toeplitz structure}, with the result that the inverse covariance matrix can be expressed in closed form by using a special case of the Gohberg-Heinig formula that is a function of strictly the forward 2-D prediction matrix polynomials. Furthermore, we present a novel method, based on a 2-D lattice algorithm, to compute the needed forward prediction matrix polynomials and discuss the difference in the so-obtained 2-D spectral estimate as compared to the one obtained by using the prediction matrix polynomials given by the Whittle-Wiggins-Robinson algorithm. Numerical simulations illustrate the clear computational gain in comparison to both the well-known classical implementation and the method recently published by Liu \textit{et al.}

6.1 Introduction

The problem of two-dimensional (2-D) high resolution spectral estimation has been widely studied in the past literature (see, e.g., [DM84, McC82, Mar87]), as well as in more recent contributions such as [LLS98, LLL98, Mar]. Applications occur in a wide variety of fields, such as geophysics, radio astronomy, biomedical engineering, sonar, and radar, to mention a few. In many of these applications, it is of key importance to obtain computationally efficient high resolution estimates, as for example it is in synthetic aperture radar (SAR) image formation and target feature extraction [DeG98, JWE+96], which are becoming increasingly important in
many civilian and military applications [Kle97]. Another important application is 2-D nuclear magnetic resonance (NMR) spectral estimation, where both resolution and computational complexity are of utmost importance [HKB+89]. Popular approaches include the 2-D Periodogram, and in the higher resolution cases, the 2-D AR and the 2-D Capon spectral estimators. A number of approaches has been suggested for efficient estimation of the 2-D AR spectrum (see, e.g., [ML89, TES89, Mar87]), whereas only limited efforts have been made to simplify the 2-D Capon estimator [LLL98, Mar87].

In this work we present an efficient implementation of the 2-D Capon spectral estimator, which will depend on the (forward) prediction matrices and the (forward) prediction error covariance matrix. The derivation is based on a special case of the Gohberg-Heinig formula [GH74] (see also [FMKL79]) for the inverse of a Block-Toeplitz matrix. This closed form expression of the inverse covariance matrix enables significant computational reduction to form the 2-D Capon spectral estimator due to the highly structured problem formulation, and the algorithm can be seen as a 2-D extension of the algorithm derived by Musicus [Mus85] for the 1-D case. The final form of the algorithm was proposed, although the algorithm was neither specified nor derived, in [Mar87]. Here, we present the full derivation of the algorithm as well as a complexity comparison with the standard 2-D Capon approach as well as with the recently published implementation by Liu et al [LLL98].

Furthermore, we present a novel 2-D lattice algorithm to estimate the needed (forward) prediction matrices and the (forward) prediction error covariance matrix. The presented algorithm is an improvement over the previous attempts to extend the linear prediction lattice-based 1-D techniques to two dimensions [Str77, ML89, TES89, KR93, Gup94]. Following the 1-D Burg recursion procedure [Bur75], Strand [Str77] extended the Whittle-Wiggins-Robinson Algorithm (WWRA) [Whi63, WR65] to compute the AR parameters directly from the multichannel data without first computing the autocorrelation sequence. Therrien and El-Shaar [TES89] used this technique to compute the 2-D AR spectral estimates directly from the data by treating the 2-D signal as being made of strips of multichannel signals. In [KR93], Kuduvalli and Rangayyan further simplified Therrien and El-Shaar’s algorithm by forcing the structure of the 2-D autocorrelation matrices on the multichannel version of the Burg algorithm. A similar approach was used by McGuffin and Liu in [ML89].

These approaches have all failed to create a proper extension of the Burg algorithm due to the failure to recognize the correct definitions for the forward and backward 2-D linear prediction errors. In this chapter, we
provide proper definitions and develop the appropriate fast computational algorithm that, from 2-D data sample sets, estimates 2-D forward and backward linear prediction parameter matrices. In the 1-D case, the algorithm will reduce to the well-known Burg algorithm. In the numerical section, we compare the computational complexity of the 2-D lattice algorithm with alternative methods for estimating the prediction error matrices. We also compare the relative quality of the spectral estimates obtained from these estimates. It is found that the spectral estimates obtained by using the new 2-D lattice algorithm are noticeably better than the ones based on using the WWRA with estimates of the 2-D covariance matrix.

The chapter is organized as follows. In Section 6.2 we review the classical 2-D Capon spectral estimator. Next, we describe the suggested efficient algorithm in Section 6.3. In Section 6.4 we review the extended Yule-Walker equations, and in Section 6.5 we introduce the novel 2-D lattice algorithm. Section 6.6 contains our numerical examples and, finally, Section 6.7 contains our conclusions.

6.2 Problem Formulation

In this section, we will briefly review the well-known 2-D Capon spectral estimation method and formulate the problem of interest. The Capon spectral estimator was, unlike many other spectral estimation techniques, originally proposed directly in a multi-dimensional setting as an array-processing technique [Cap69, Cap83]. It is a non-parametric adaptive matched-filterbank approach [SJL98, LLS98] (see also Chapter 2), and follows two main steps¹

a) Pass the data through a 2-D bandpass filter with varying center frequencies $(\omega_1, \omega_2)$, and

b) estimate the power of the filter output, $S_{\omega_1, \omega_2}$, for all $\omega_1 \in [0, 2\pi), \omega_2 \in [0, 2\pi)$ of interest from the filtered data.

Let the $N_1 \times N_2$ data matrix $Z$ be a part of a 2-D stationary discrete data sequence, and assume that the bandpass filter used, $H_{\omega_1, \omega_2}$, is an $(L_1 + 1) \times (L_2 + 1)$-tap 2-D finite impulse response (FIR) filter. The filterlengths $L_1$ and $L_2$ are parameters specified by the user.

¹Note that the estimated power in step (b) will need to be scaled with the bandwidth of the filter to yield an appropriate estimate of the power spectral density (PSD) (see Section 1.2.2 for a further discussion on this issue). In the current chapter, we will not delve further into this aspect.
Define the \((L_1 + 1)(L_2 + 1) \times 1\) filter vector

\[
h_{\omega_1, \omega_2} \triangleq \text{vec}(H_{\omega_1, \omega_2}), \tag{6.2.1}
\]

where \(\text{vec}(\cdot)\) denotes the operation of stacking the columns of a matrix on top of each other. Form the \((L_1 + 1) \times (L_2 + 1)\) submatrices

\[
Y_{t, s} = \begin{bmatrix}
Z_{t+L_1,s+L_2} & \cdots & Z_{t,s+L_2} \\
\vdots & \ddots & \vdots \\
Z_{t+L_1,s} & \cdots & Z_{t,s}
\end{bmatrix} \tag{6.2.2}
\]

for \(t = 0, \ldots, M_1, s = 0, \ldots, M_2\), where \(M_1 = N_1 - L_1 - 1, M_2 = N_2 - L_2 - 1\), and define the \((L_1 + 1)(L_2 + 1) \times 1\) snapshot vector

\[
y_{t, s} = \text{vec}(Y_{t, s}). \tag{6.2.3}
\]

Furthermore, define the covariance matrix, \(R\), as

\[
R \triangleq \mathbb{E}\{y_{t, s} y_{t, s}^*\} = \begin{bmatrix}
R_0 & R_1 & \cdots & R_{L_1} \\
R_1^* & R_0 & \ddots & \vdots \\
\vdots & \ddots & \ddots & R_1 \\
R_{L_1}^* & \cdots & R_1^* & R_0
\end{bmatrix}, \tag{6.2.4}
\]

where \(\mathbb{E}\{\cdot\}\) and \((\cdot)^*\) denote the expectation and the complex conjugate transpose, and where the block matrices \(R_k\) are

\[
R_k = \begin{bmatrix}
\tau_{k,0} & \tau_{k,1} & \cdots & \tau_{k,L_2} \\
\tau_{k,1}^* & \tau_{k,0} & \ddots & \vdots \\
\vdots & \ddots & \ddots & \tau_{k,1} \\
\tau_{k,L_2}^* & \cdots & \tau_{k,1}^* & \tau_{k,0}
\end{bmatrix}, \tag{6.2.5}
\]

where

\[
\tau_{k,l} \triangleq \mathbb{E}\{Z_{t+k,s+l} Z_{t,s}^*\} = r_{-k,-l}^*. \tag{6.2.6}
\]

It can be seen from (6.2.4) and (6.2.5) that the \((L_1 + 1)(L_2 + 1) \times (L_1 + 1) (L_2 + 1)\) covariance matrix \(R\) has a (Hermitian) Toeplitz-Block-Toeplitz structure.

The adaptive Capon bandpass filter is designed to minimize the power of the filter output, as well as pass the frequencies \((\omega_1, \omega_2)\) without any attenuation, i.e., to satisfy, for each \((\omega_1, \omega_2)\),

\[
\min_h h^*_{\omega_1, \omega_2} R h_{\omega_1, \omega_2} \quad \text{subject to} \quad h^*_{\omega_1, \omega_2} a_{\omega_1, \omega_2} = 1, \tag{6.2.7}
\]
where $\mathbf{a}_{\omega_1,\omega_2}$ is the 2-D Fourier vector, defined as

$$
\mathbf{a}_{\omega_1,\omega_2} \triangleq \mathbf{a}_{\omega_1} \otimes \mathbf{a}_{\omega_2}
$$

and

$$
\mathbf{a}_{\omega_k} \triangleq \begin{bmatrix}
1 & e^{-i\omega_k} & \ldots & e^{-iM_k\omega_k}
\end{bmatrix}^T.
$$

(6.2.8)

Hereafter, $\otimes$ and $(\cdot)^T$ denote the Kronecker product and the transpose, respectively. When the filter minimizing (6.2.7) is inserted into the expression for the power of the filter output, for the bandpass filter centered at the frequencies of interest, $(\omega_1, \omega_2)$, i.e.,

$$
\mathbf{S}_{\omega_1,\omega_2} = \mathbf{h}^*_{\omega_1,\omega_2} \mathbf{R} \mathbf{h}_{\omega_1,\omega_2},
$$

(6.2.10)

the (Power Spectrum) Capon spectral estimate at $(\omega_1, \omega_2)$ is obtained as

$$
\hat{\mathbf{S}}_{\omega_1,\omega_2} = \frac{1}{\mathbf{a}^*_{\omega_1,\omega_2} \mathbf{R}^{-1} \mathbf{a}_{\omega_1,\omega_2}}.
$$

(6.2.11)

The problem of interest is to compute (6.2.11) efficiently. We note that, the primary computational burden to evaluate (6.2.11) is not, as it might first seem, that associated with the inverse of the covariance matrix $\mathbf{R}$. If that was the case, an efficient algorithm for the inversion of a Toeplitz-Block-Toeplitz matrix could have been used [WK83, KCM84]. Rather, it is the computation of (6.2.11) over all frequencies that is normally more time consuming. In the following section, we propose an efficient way to do this using the 2-D Fast Fourier Transform.

### 6.3 Proposed Algorithm

By making use of the fact that the inverse of a Block-Toeplitz matrix can be formulated in a closed-form expression by using the Gohberg-Heinig formula as well as the relationship between the forward and backward prediction matrix polynomials that holds for Toeplitz-Block-Toeplitz covariance matrices, we are now ready to state the first main result of this chapter. For the reader’s convenience, the Gohberg-Heinig formula is summarized in Appendix 6.A.

**Theorem 6.1**

*The denominator of the 2-D Capon spectral estimator*

$$
\mathbf{f}_{\omega_1,\omega_2} \triangleq \mathbf{a}^*_{\omega_1,\omega_2} \mathbf{R}^{-1} \mathbf{a}_{\omega_1,\omega_2},
$$

(6.3.12)

can be equivalently expressed as

$$
\mathbf{f}_{\omega_1,\omega_2} = \sum_{s=-L_1}^{L_1} \sum_{p=-L_2}^{L_2} \theta(s, p) e^{-i\omega_1 s} e^{-i\omega_2 p}
$$

(6.3.13)
where \( \theta(s, p) = \bar{\theta}(-s, -p) \),

\[
\theta(s, p) = \sum_{k=0}^{L_1-s} (L_1 + 1 - 2k - s) \Psi(s, p, k) \tag{6.3.14}
\]

\[
\Psi(s, p, k) = \sum_{l=\max(0,-p)}^{\min(L_2-p,L_2)} [A_k Q^{-1} A_{k+s}]_{l,l+p} \tag{6.3.15}
\]

and where \( \{A_k\} \) and \( Q \) denote the (forward) prediction matrices and the (forward) prediction error covariance matrix as defined in (6.4.17), with \( R \) being given by (6.2.4). Here, \( \bar{x} \) and \( |X|_{i,j} \) denote the complex conjugate of \( x \) and the \( i,j \)-th entry of the matrix \( X \).

**Proof:** See Appendix 6.B.

---

![Diagram](image.png)

**Figure 6.1:** The support of the \( \theta \) parameters

It is worth noting that the expression for \( f_{\omega_1,\omega_2} \) in (6.3.13) can be efficiently computed by making use of the periodic properties of the 2-D FFT algorithm. Denote the support of the \( \theta \) parameters with \( \Omega_k \) as is shown in Figure 6.1, i.e.,

\[
\Omega_1 = \begin{bmatrix}
\theta_{L_1,0} & \ldots & \theta_{L_1,L_2} \\
\vdots & \ddots & \vdots \\
\theta_{0,0} & \ldots & \theta_{0,L_2}
\end{bmatrix} \quad \Omega_2 = \begin{bmatrix}
\theta_{-1,0} & \ldots & \theta_{-1,L_2} \\
\vdots & \ddots & \vdots \\
\theta_{-L_1,0} & \ldots & \theta_{-L_1,L_2}
\end{bmatrix}
\]

\[
\Omega_3 = \begin{bmatrix}
\theta_{-1,-L_2} & \ldots & \theta_{-1,-1} \\
\vdots & \ddots & \vdots \\
\theta_{-L_1,-L_2} & \ldots & \theta_{-L_1,-1}
\end{bmatrix} \quad \Omega_4 = \begin{bmatrix}
\theta_{L_1,-L_2} & \ldots & \theta_{L_1,-1} \\
\vdots & \ddots & \vdots \\
\theta_{0,-L_2} & \ldots & \theta_{0,-1}
\end{bmatrix}
\]

Next, introduce the \( N_{\omega_1} \times N_{\omega_2} \tilde{\Omega} \), where \( N_{\omega_1} \) and \( N_{\omega_2} \) denote the number
of frequencies for which the spectrum should be computed, and let

$$\Omega = \begin{bmatrix} \Omega_2 & 0 & \Omega_3 \\ 0 & 0 & 0 \\ \Omega_1 & 0 & \Omega_4 \end{bmatrix}. \quad (6.3.16)$$

Note that the different zero matrices, $\mathbf{0}$, in (6.3.16) have different dimensions. A positive real-valued $f_{\omega_1, \omega_2}$ can then be efficiently computed as the $N_{\omega_1} \times N_{\omega_2}$ 2-D FFT of $\Omega$.

It is also worthwhile to note that, as the algorithm in Theorem 6.1 requires the computation of the $(L_2 + 1) \times (L_2 + 1)$ $\{\mathbf{A}_i\}$ and $\mathbf{Q}$ matrices from the data, as well as the matrix inversion of the $\mathbf{Q}$ matrix in (6.3.15), it may be computationally prohibitive to apply the algorithm directly to data matrices with very large dimensions. One can then proceed in a different way as was suggested in [PL98]. Break up each frequency domain image (obtained by taking the 2-D FFT of the time domain data) into overlapping chips of size $N_{s_1} \times N_{s_2}$ (where $N_{s_1}$ and $N_{s_2}$ are smaller than the original data dimensions), as shown in Figure 6.2. Take the 2-D inverse FFT (IFFT) of the frequency domain chips to obtain the time domain chips and apply the algorithm in Theorem 6.1 to the time domain chips. This approach will not produce any mosaicing or tiling effect.

As (6.3.15) depends on the (forward) prediction matrices and the (forward) prediction error covariance matrix, the problem of computing these still remains. In the following two sections we first review the extended Yule-Walker solution and then introduce the novel 2-D lattice algorithm for computing $\{\mathbf{A}_i\}$ and $\mathbf{Q}$ from the data.
6.4 The Yule-Walker Equations

The extended Yule-Walker equations can be expressed as [Mar87, SS89]

\[
\begin{bmatrix}
I & A_1 & \cdots & A_n \\
B_n & \cdots & B_1 & I
\end{bmatrix}
\begin{bmatrix}
R
\end{bmatrix}
\begin{bmatrix}
Q & 0 & \cdots & 0 \\
0 & \cdots & 0 & S
\end{bmatrix}
\]

(6.4.17)

where \( R \) is defined as in (6.2.4), with \( n = L_1 \). In the case that \( R \) has a
Toeplitz-Block-Toeplitz structure, it is known that the backward prediction
matrices and the backward prediction error covariance matrix can be found
as [Mar87, ML89, KH87]

\[
B_i = J\hat{A}_iJ
\]

(6.4.18)

\[
S = JQJ
\]

(6.4.19)

where \( J \) denotes the so-called exchange matrix, whose anti-diagonal
elements are ones and all other elements are zero, i.e.,

\[
J = \begin{bmatrix}
0 & \cdots & 0 & 1 \\
0 & \cdots & 1 & 0 \\
\vdots & \ddots & \ddots & \vdots \\
1 & 0 & \cdots & 0
\end{bmatrix}.
\]

(6.4.20)

In fact, the above identities will hold if \( R \) is simply persymmetric [Mar87],
i.e.,

\[
R = JR^TJ.
\]

(6.4.21)

Several approaches to solve the extended Yule-Walker equations in (6.4.17),
making use of the identities in (6.4.18) and (6.4.19), have been proposed
in the literature (see, e.g., [Mar87, ML89, KH87]), improving computationally
on the Whittle-Wiggins-Robinson algorithm (WWRA). The resulting
algorithms reduce the computational burden by approximately one-half as
compared to the WWRA. However, this will hold only for larger matrices,
whereas for smaller matrices of the dimensions considered in this paper it
might still be more efficient to use the WWRA. For this reason, we have in
the numerical section only used the WWRA. For easy reference the WWRA
is summarized in Appendix 6.C.

As a first step in using any of these algorithms, the covariance matrix,
\( R \), needs to be estimated. Just as in the 1-D case, there are several different
approaches to do so, which will yield different estimates of the prediction
matrices.
In this chapter we will consider and compare the following different approaches of estimating the covariance matrix:

- The Toeplitz-Block-Toeplitz estimate.
- The forward-backward (FB) Block-Toeplitz estimate.
- The outer product estimate.

For completeness, we will in the following subsections quickly review the different estimates.

### 6.4.1 The Toeplitz-Block-Toeplitz Estimate

The Toeplitz-Block-Toeplitz covariance matrix estimate is obtained by building the full matrix from the autocorrelation estimates, $\hat{r}_{k,l}$, as defined in (6.2.6). These are obtained as

$$\hat{r}_{k,l} = \begin{cases} 
\frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1-k} \sum_{n=0}^{N_2-1-l} Z_{m+k,n+l} Z_{m,n}^* & \text{for } k \geq 0, l \geq 0 \\
\frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1-k} \sum_{n=-l}^{N_2-1} Z_{m+k,n+l} Z_{m,n}^* & \text{for } k \geq 0, l < 0 \\
\hat{r}_{-k,-l}^* & \text{for } k \leq 0, \text{any } l 
\end{cases}$$

over a lag range of $|k| \leq L_1$ and $|l| \leq L_2$.

### 6.4.2 The FB Block-Toeplitz Estimate

The FB Block-Toeplitz covariance matrix is obtained by estimating the forward-backward block sample covariance matrices, $\hat{R}_k^{fb}$, and then proceeding by building the full covariance matrix using its Block-Toeplitz structure to obtain $\hat{R}^{fb}$. The $k$:th block forward covariance matrix is estimated as (for $k = 0, \ldots, L_1$)

$$\hat{R}_k^f = \frac{1}{N_1 N_2} \sum_{t=0}^{N_1-1-k} \sum_{s=0}^{N_2-1-L_2} \begin{bmatrix} Z_{t+k,s} \\ \vdots \\ Z_{t+k,L_2} \end{bmatrix} \begin{bmatrix} Z_{t,s+L_2}^* & \cdots & Z_{t,s}^* \end{bmatrix}.$$

The forward-backward sample covariance matrix is obtained as

$$\hat{R}_k^{fb} = \frac{1}{2} \left( \hat{R}_k^f + \text{J} \left( \hat{R}_k^f \right)^T \text{J} \right). \quad (6.4.22)$$
Note that, although $\hat{R}_k^{f,b}$ in (6.4.22) is not Toeplitz, it is persymmetric which also makes $\hat{R}_k^{f,b}$ persymmetric. Since $R_k$ is persymmetric, one would expect that $\hat{R}_k^{f,b}$ is a better estimate of $R_k$ than the non-persymmetric $R_k^{f}$. We refer the reader to [JS99] for an analysis and a comparison between the use of the forward-only and the forward-backward sample covariance matrices for the 1-D Capon spectral estimator. Similar conclusions are expected to hold in the 2-D case considered here.

6.4.3 The Outer Product Estimate

The outer product sample covariance matrix is obtained by estimating the covariance matrix without taking its internal structure into account, i.e.,

$$
\hat{R} = \frac{1}{M_1 M_2} \sum_{k=0}^{M_1} \sum_{l=0}^{M_2} \text{vec}(Y_{k,l}) \text{vec}(Y_{k,l})^*.
$$

6.5 The 2-D Lattice Algorithm

In this section we present a novel 2-D lattice algorithm, that obtains $\{A_i\}$ and $Q$ directly from the data $\{Z_{k,l}\}$. This algorithm, which does not depend on an estimate of the covariance matrix $R$, is an improvement over previous attempts to extend the linear prediction lattice-based 1-D techniques to two dimensions [Str77, ML89, KR93, Gup94]. In the 1-D case, the algorithm will reduce to the well-known Burg algorithm [Bur75]. The algorithm has also been published as [Mar99].

The generalized $(L_2 + 1) \times (N_2 - L_2)$ 2-D forward and backward linear prediction errors for a $n$th order model at sample index $k$, for $n \leq k \leq N_1 - 1$, are defined as

$$
e^{f}_{n}(k) \triangleq X_k + \sum_{l=1}^{n} A_{l}^{(n)}X_{k-l} \quad (6.5.23)
$$

$$
e^{b}_{n}(k) \triangleq X_{k-n} + \sum_{l=1}^{n} (J\tilde{A}_{l}^{(n)}J)X_{k-n+l}, \quad (6.5.24)
$$

where $\{A_i\}$ are the $(L_2 + 1) \times (L_2 + 1)$ forward prediction matrices defined in (6.4.17), and where the $(L_2 + 1) \times (N_2 - L_2)$ Toeplitz data matrix at index $k$ is defined as

$$
X_k \triangleq \begin{bmatrix}
Z_{k,L_2} & \ldots & Z_{k,2L_2} & \ldots & Z_{k,N_2-1} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
Z_{k,0} & \ldots & Z_{k,L_2} & \ldots & Z_{k,N_2-L_2-1}
\end{bmatrix}, \quad (6.5.25)
$$
where the dimensional constraint $N_2 - L_2 \geq L_2 + 1$ enables the product $X_k X_k^*$ to be non-singular. Note that at order $n = 0$, $e^f_0(k) = e^b_0(k) = X_k$.

Define the estimated 2-D forward and backward linear prediction squared errors as

$$
P^f_n \triangleq \frac{1}{N_1 - n} \sum_{k=n}^{N_1-1} e^f_n(k) \left( e^f_n(k) \right)^* \quad (6.5.26)
$$

$$
P^b_n \triangleq \frac{1}{N_1 - n} \sum_{k=n}^{N_1-1} e^b_n(k) \left( e^b_n(k) \right)^* \quad (6.5.27)
$$

A recursive 2-D lattice relationship may be developed between the 2-D forward and backward linear prediction matrix errors by using the order update of the WWRA, for $k = 1, \ldots, n$, (cf. (6.1.1) and (6.4.18))

$$
A^{(n+1)}_k = A^{(n)}_k + K^{n+1}_n \left( J \tilde{A}^{(n)}_{n+1-k} J \right) \quad (6.5.28)
$$

where $K^{n+1}_n = A^{(n+1)}_{n+1}$, the so-called 2-D reflection coefficient matrix [Mak75], is the gain coefficient matrix of the 2-D generalized lattice filter. By substituting (6.5.28) into (6.5.23) and (6.5.24), we obtain

$$
e^f_{n+1}(k) = e^f_n(k) + K^{n+1}_n e^b_n(k-1)
$$

$$
e^b_{n+1}(k) = e^b_n(k-1) + (J K^{n+1}_n J) e^f_n(k). \quad (6.5.29)
$$

Making use of (6.4.17), as well as the identities (6.4.18) and (6.4.19), we note that

$$
E \left( \Phi^f_n \right) = \left[ \begin{array}{cccc} I & A^{(n)}_1 & \cdots & A^{(n)}_n \end{array} \right] R \left[ \begin{array}{cccc} I & A^{(n)}_1 & \cdots & A^{(n)}_n \end{array} \right]^* = E \left( \Phi^{b}_{n} J \right). \quad (6.5.30)
$$

where $R$ is given in (6.2.4), with $n = L_1$. From (6.5.30), it can be seen that both $E(\Phi^f_n)$ and $E(\Phi^{b}_{n} J)$ will have the prediction error autocovariances along their main diagonals. Thus, a least squares estimate of $K^{n+1}$ can be obtained by minimizing

$$
\text{tr} \left\{ \Phi^{f+b}_{n+1} \right\}, \quad (6.5.31)
$$

where

$$
\Phi^{f+b}_{n+1} \triangleq \Phi^f_n + J \tilde{\Phi}^b_n J, \quad (6.5.32)
$$

as this will be equivalent with minimizing the sum of the prediction error autocovariances. This should be compared with the minimization criteria used in [ML89, KR93], where $\text{tr} \left( \Phi^f_n + \Phi^b_n \right)$ is instead minimized.
By substituting (6.5.29) into (6.5.26) and (6.5.27), (6.5.32) may be expressed as

$$ P_{n+1}^{f+b} = P_n^{f+b} + K_{n+1}^{*} + \hat{\Delta}_n K_{n+1}^{*} + K_{n+1}^{*} \left[ J P_n^{f+b} J \right] K_{n+1}^{*} $$

$$ = P_n^{f+b} - \hat{\Delta}_n \left[ J P_n^{f+b} J \right]^{-1} \hat{\Delta}_n + C_{n+1} \left[ J P_n^{f+b} J \right] C_{n+1}^{*} $$

(6.5.33)

where

$$ C_{n+1} \triangleq K_{n+1}^{*} + \hat{\Delta}_n \left[ J P_n^{f+b} J \right]^{-1} $$

(6.5.34)

$$ \hat{\Delta}_n \triangleq P_n^{f+b} + J \left( P_n^{f+b} \right)^T J $$

(6.5.35)

$$ P_n^{f+b} \triangleq \frac{1}{N_1 - n} \sum_{k=n}^{N_1-1} e_n^f(k) \left( e_n^b(k - 1) \right)^* $$

(6.5.36)

Here, $P_n^{f+b}$ is an estimate of the cross-correlation between the generalized forward and backward 2-D linear prediction errors, and (the persymmetric) $\hat{\Delta}_n$ is an approximation for the partial correlation matrix of the 2-D Yule-Walker solution.

To minimize the trace in (6.5.31), with respect to the single unknown matrix $K_{n+1}$, we note that only the third term in (6.5.33) depends on $K_{n+1}$. Because this third term is a quadratic matrix product that can only contribute positive values to the diagonal elements of $P_n^{f+b}$, the only choice that will minimize the trace is $C_{n+1} = 0$, or

$$ K_{n+1} = -\hat{\Delta}_n \left[ J P_n^{f+b} J \right]^{-1} $$

$$ = - \left[ P_n^{f+b} + J \left( P_n^{f+b} \right)^T J \right] \left[ P_n^{f+b} + J P_n^{f+b} J \right]^{-1}, $$

(6.5.37)

The sum of forward and backward 2-D squared errors that minimizes the trace will then have the following recursive relationship between orders

$$ P_{n+1}^{f+b} = \left( P_{n+1}^{f+b} \right)^* $$

$$ = P_n^{f+b} + K_{n+1} \hat{\Delta}_n $$

$$ = P_n^{f+b} - \hat{\Delta}_n \left[ J P_n^{f+b} J \right]^{-1} \hat{\Delta}_n. $$

(6.5.38)

By finally combining (6.5.26), (6.5.27), (6.5.36), (6.5.37), (6.5.38), (6.5.28) and (6.5.29), in that order, the algorithm is obtained. This algorithm constitutes the chapter’s second main contribution.
6.6 Numerical Examples

We first study the computational complexity of the different methods. In our first example, the data matrix has been generated as the sum of four 2-D complex sinusoids (cisoids) corrupted by additive complex Gaussian white noise. The computational complexity is evaluated as the data matrix size, \( N_1 = N_2 \), varies. Figure 6.3(a) illustrates the computational complexity of the proposed spectral estimators for varying data matrix dimensions, as compared to the direct solution for the classical 2-D Capon spectral estimator, as given in (6.2.11), and the method in [LLL98]. The figure shows the number of floating point operations (flops) as measured by MATLAB for the calculation of the 2-D spectrum. As a comparison, Figure 6.3(b) shows the computational requirements for computing the outer product sample covariance matrix for the classical 2-D Capon spectral estimator and the estimator in [LLL98], and the \( \{A_i\} \) and \( Q \) matrices using the different approaches for all the other approaches based on Theorem 6.1. We note that there is a clear computational gain in using any of the spectral estimates based on Theorem 6.1 as compared to both the classical approach and the algorithm presented in [LLL98]. In the example, the filter lengths were \( L_1 = L_2 = N_1/4 \), and the spectrum is zeropadded to length \( N_{\omega_1} = N_{\omega_2} = 4N_1 \) (which means that the spectrum is evaluated using a \( 4N_1 \times 4N_1 \)-point 2-D FFT).

We will now proceed by studying the spectral resolutions achieved by the different methods. First, we use a \( 32 \times 32 \) data matrix that consists of a sum of four cisoids corrupted by additive complex Gaussian white noise. The cisoids all have unit amplitude, a phase of \( \pi/4 \), and are located at \( f = (-0.25, 0.25), (0, 0), (-0.25, 0), (0, -0.25) \), as is shown in Figure 6.4(a). Figure 6.4(b) and Figures 6.5(a)–(d) illustrate the resulting resolutions obtained by using the classical 2-D Capon estimate, the Toeplitz-Block-Toeplitz 2-D Capon estimate, the FB Block-Toeplitz 2-D Capon estimate, the 2-D lattice Capon estimate and the 2-D Capon estimate of [LLL98], respectively.

We note that there are significant differences between the different estimates. First, note that the 2-D classical Capon spectral estimate, based on the outer-product covariance matrix estimate, as shown in Figure 6.4(b), is seen to have a high frequency resolution but to provide a slightly biased amplitude estimates. This is well in accordance with the known fact that Capon will yield an amplitude estimate that is biased downwards [LLS98, SJL98]. Next, the spectral estimate based on the Toeplitz-Block-Toeplitz estimated covariance matrix, as shown in Figure 6.5(a), is seen to give fairly wide peak estimates. Figure 6.5(b) shows that the spectrum
Figure 6.3: Computational complexity vs data matrix size ($N_1 = N_2$)
(a) Computation of the spectral estimates. (b) Computation of the outer product covariance matrix for the classical and the Liu et al. Capon spectral estimators, and the $A_i$ matrices for the others.

Based on FB Block-Toeplitz estimated covariance yields high resolution for one of the frequencies\footnote{This is due to the “row-averaging” in (6.2.3). See [OBP97] for a further discussion on this phenomena, as well as [Mar87, JC79] for suggested remedies.} although with significantly worse amplitude estimates. The Capon estimate of [LLL98] is shown in Figure 6.5(c), and the
spectral estimate based on the 2-D lattice estimates is shown in Figure 6.5(d). The methods are seen to have similar resolution.

We conclude that the spectral estimate based on the $A_i$ matrices as given by the 2-D lattice algorithm seems to give a better spectral estimate than the estimate given by the extended Yule-Walker solutions. This is well in accordance with the 1-D case, where the Yule-Walker algorithm’s $A_i$ estimates will result in lower resolution than estimates obtained by the Burg algorithm [UB75]. In the example, the filter lengths were $L_1 = L_2 = N_1/4$, and the spectrum is zero-padded to length $N_{\omega_1} = N_{\omega_2} = 4N_1$.

![Figure 6.4](image_url)

**Figure 6.4:** Illustration of the resolution and accuracy for the different spectral estimators. The estimates are plotted for fractions of the sampling frequency. (a) The true spectrum. (b) The classical 2-D Capon estimate.

Finally, we consider synthetic aperture radar (SAR) imaging of a simulated MIG-25 airplane. The $32 \times 32$ data matrix was provided by the Naval Research Laboratory. Figures 6.6(a)–(d) show the $128 \times 128$ SAR images as obtained by the different methods, using $L_1 = L_2 = 15$. We note that the proposed lattice-based 2-D Capon spectral estimate seems to give the clearest image.
Figure 6.5: Illustration of the resolution and accuracy for the different spectral estimators. The estimates are plotted for fractions of the sampling frequency. (a) The Toeplitz-Block-Toeplitz 2-D Capon estimate, (b) The FB Block-Toeplitz 2-D Capon estimate, (c) The 2-D Capon estimate of [LLL98], (d) The 2-D lattice Capon estimate.
Figure 6.6: Illustration of the resolution and accuracy for the different spectral estimators. (a) The 2-D Periodogram. (b) The Toeplitz-Block-Toeplitz 2-D Capon estimate. (c) The 2-D Capon estimate of [LLL98]. (d) The 2-D lattice Capon estimate.
6.7 Conclusions

In this work we have presented an efficient implementation of the 2-D Capon spectral estimator. The derivation is based on the Gohberg-Heinig formula for the inverse of a Block-Toeplitz matrix. The so-obtained closed form expression of the inverse covariance matrix enables further simplifications of the 2-D Capon spectral estimator due to the highly structured problem formulation and the algorithm can be seen as a 2-D extension of the algorithm derived by Musicus [Mus85] for the 1-D case. Our numerical simulations indicate a significant decrease in computational complexity as compared to the classical approach as well as the approach recently proposed by Liu et al [LLL98], especially for larger matrix sizes.

Furthermore, we have presented a novel 2-D lattice algorithm to estimate the forward prediction matrices as well as the forward prediction error covariance matrix. The so-obtained estimates are found to produce a spectral estimate with high frequency resolution. Our numerical examples illustrate the improved resolution.
6.A The Gohberg-Heinig formula

The Gohberg-Heinig formula, presented in [GH74] (see also [FMKL79]), is an extension of the well-known Gohberg-Semencul formula [GS72] (see also, e.g., [SM97]) to the Block-Toeplitz case. The formula is summarized below.

**Theorem 6.2 (The Gohberg-Heinig formula)**

Let \( R \) be the \((L_1 + 1)(L_2 + 1) \times (L_1 + 1)(L_2 + 1)\) non-singular Hermitian Block-Toeplitz matrix in (6.2.4). Then it holds that

\[
R^{-1} = L^*_A \, D(Q^{-1}) \, L_A - L^*_B \, D(S^{-1}) \, L_B \tag{6.A.1}
\]

where \( D(Q^{-1}) \) is a block diagonal matrix with the matrix \( Q^{-1} \) along its diagonal,

\[
L_A = \begin{bmatrix}
I & A_1 & \cdots & A_{L_1} \\
0 & I & \ddots & \vdots \\
\vdots & \ddots & \ddots & A_1 \\
0 & \cdots & 0 & I
\end{bmatrix},
\]

\[
L_B = \begin{bmatrix}
0 & B_{L_1} & \cdots & B_1 \\
0 & 0 & \ddots & \vdots \\
\vdots & \ddots & \ddots & B_{L_1} \\
0 & \cdots & 0 & 0
\end{bmatrix},
\]

and where \( \{A_i\}, \{B_i\}, Q \) and \( S \) are the \((L_2 + 1) \times (L_2 + 1)\) forward and backward prediction matrices and the forward and backward prediction error covariance matrices of order \( L_1 \) as defined in (6.4.17).

**Proof:** See [GH74, FMKL79]. \( \Box \)

Note that, by making use of the identities (6.4.18) and (6.4.19), the Gohberg-Heinig formula can, for a Toeplitz-Block-Toeplitz matrix, be expressed in terms of only the forward prediction matrices and the forward prediction error covariance matrix as

\[
R^{-1} = L^*_A \, D(Q^{-1}) \, L_A - L^*_A \, D((JQJ)^{-1}) \, L_A
\]

where

\[
L_A = \begin{bmatrix}
0 & (J\tilde{A}_{L_1}J) & \cdots & (J\tilde{A}_1J) \\
0 & 0 & \ddots & \vdots \\
\vdots & \ddots & \ddots & (J\tilde{A}_1J) \\
0 & \cdots & 0 & 0
\end{bmatrix}.
\]
6.B Proof of Theorem 6.1

Making use of the Gohberg-Heinig formula, as stated in Appendix 6.A, we rewrite $f_{\omega_1, \omega_2}$ as

$$f_{\omega_1, \omega_2} = \sum_{p=0}^{L_1} \sum_{k=0}^{L_1} \sum_{l=0}^{L_1} a_{\omega_2}^* (A_{k-p}^* Q^{-1} A_{l-p} - B_{L_1+1+p-k}^* S^{-1} B_{L_1+1+p-l} a_{\omega_2} e^{i\omega_1 (k-l)})$$

$$= \sum_{p=0}^{L_1} \sum_{l=0}^{L_1} \sum_{s=l-L_1}^{L_1} a_{\omega_2}^* (A_{l-p-s}^* Q^{-1} A_{l-p} - B_{L_1+1+p-l-s}^* S^{-1} B_{L_1+1+p-l} a_{\omega_2} e^{-i\omega_1 s}), \quad (6.B.1)$$

where the last equality has been obtained by substitution $s = l - k$. Next, make substitution $j = l - p$ in (6.B.1) to obtain

$$f_{\omega_1, \omega_2} = \sum_{l=0}^{L_1} \sum_{s=l-L_1}^{L_1} \sum_{j=l-L_1}^{L_1} a_{\omega_2}^* (A_{j-s}^* Q^{-1} A_j - B_{L_1+1+s-j}^* S^{-1} B_{L_1+1+l-j} a_{\omega_2} e^{-i\omega_1 s})$$

$$= \sum_{l=0}^{L_1} \sum_{s=l-L_1}^{L_1} \sum_{j=0}^{L_1} a_{\omega_2}^* (A_{j-s}^* Q^{-1} A_j - B_{L_1+1+s-j}^* S^{-1} B_{L_1+1+l-j} a_{\omega_2} e^{-i\omega_1 s}), \quad (6.B.2)$$

where we have made use of the fact that the summand is zero for $j < 0$, and hence we can truncate the summation over $j$ to the interval $[0, l]$. Furthermore, we have made use of the fact that since $A_{j-s} = B_{L_1+1+s-j} = 0$ for $s > j$, we can extend the summation over $s$ up to $s = L_1$. We proceed by writing (6.B.2) as

$$f_{\omega_1, \omega_2} \triangleq T_1 + T_2,$$

where

$$T_1 = \sum_{s=0}^{L_1} \mu_s e^{-i\omega_1 s}$$

$$T_2 = \sum_{s=-L_1}^{L_1} \rho_{-s} e^{-i\omega_1 s}$$

and, where

$$\mu_s \triangleq \sum_{l=0}^{L_1} \sum_{j=0}^{L_1} a_{\omega_2}^* (A_{j-s}^* Q^{-1} A_j - B_{L_1+1+s-j}^* S^{-1} B_{L_1+1+l-j} a_{\omega_2} e^{-i\omega_1 s}). \quad (6.B.3)$$
As \( f_{\omega_1,\omega_2} = T_1 + T_2 \in \mathbb{R} \) for all \( \omega_1, \omega_2 \in [-\pi, \pi] \), it holds that

\[
0 = T_1 + T_2 - T_1^* - T_2^* = \sum_{k=1}^{m} (\mu_k - \rho_k^*) e^{-i\omega_1 k} + (\rho_k - \mu_k^*) e^{i\omega_1 k} = \sum_{k=1}^{m} \alpha_k z^{-k} + \alpha_k^* z^k = \sum_{k=0}^{2m} \gamma_k z^k
\]

(6.B.4)

where \( z = e^{i\omega_1} \), \( \alpha_k \triangleq \mu_k - \rho_k^* \), and

\[
\gamma_k = \begin{cases} 
-\alpha_k^* - L_1 & k = L_1 + 1, \ldots, 2L_1 \\
0 & k = L_1 \\
\alpha_k & k = 0, \ldots, L_1 - 1 
\end{cases}
\]

The polynomial in (6.B.4) has, according to the fundamental theorem of algebra \([Gau99]\), exactly \( 2L_1 \) zeros, and as (6.B.4) holds for all \( \omega_1 \in [-\pi, \pi] \), and thus providing an infinite number of roots, it must hold that \( \gamma_k = 0 \). Thus,

\[
\mu_k = \rho_k^* 
\]

which concludes that

\[
f_{\omega_1,\omega_2} = \sum_{s=-M_1}^{L_1} \mu_s e^{-i\omega_1 s}.
\]

(6.B.5)

Next, we proceed by noting that the summand in (6.B.3) does not depend on \( l \) and can thus be written as

\[
\mu_s = \sum_{j=0}^{L_1} (L_1 + 1 - j) a_{\omega_2}^* (A_{j-s}^* Q^{-1} A_j - B_{L_1+1-s-j}^* S^{-1} B_{L_1+1-j}) a_{\omega_2} - \sum_{k=-s}^{L_1-1-s} (L_1 + 1 - k - s) a_{\omega_2}^* (A_{k}^* Q^{-1} A_{k+s}) a_{\omega_2} - \sum_{k=-s}^{L_1-1-s} (L_1 + 1 - k - s) a_{\omega_2}^* (B_{L_1+1-k}^* S^{-1} B_{L_1+1-k-s}) a_{\omega_2} = \sum_{k=0}^{L_1-1-s} (L_1 + 1 - k - s) a_{\omega_2}^* (A_{k}^* Q^{-1} A_{k+s}) a_{\omega_2} - \sum_{k=0}^{L_1-1-s} (L_1 + 1 - k - s) a_{\omega_2}^* (B_{L_1+1-k}^* S^{-1} B_{L_1+1-k-s}) a_{\omega_2}
\]

(6.B.5)
\[
- \sum_{l=1}^{L_1 + 1} l a_{\omega_2}^* (B_{l+s}^* S^{-1} B_l) a_{\omega_2} \\
= \sum_{k=0}^{L_1 - s} (L_1 + 1 - k - s) a_{\omega_2}^* (A_k^* Q^{-1} A_{k+s}) a_{\omega_2} \\
- k a_{\omega_2}^* (B_{k+s}^* S^{-1} B_k) a_{\omega_2} 
\]  
(6.B.6)

where the second equality has been obtained by substituting \( j = k + s \). Furthermore, the third equation is obtained by again noting that \( A_k = 0 \) for \( k < 0 \), as well as substituting \( l = L_1 + 1 - k - s \) in the second summation, and the fourth equality is obtained by noting that \( B_k = 0 \) for \( k > L_1 \). Next, we make use of (6.4.18) and (6.4.19) to rewrite

\[
a_{\omega_2}^* (B_{k+s}^* S^{-1} B_k) a_{\omega_2} = a_{\omega_2}^* (J A_k^* T A_k^* Q^{-1} A_{k+s}) a_{\omega_2} \\
= a_{\omega_2}^* (J A_k^* Q^{-1} A_{k+s})^T J a_{\omega_2} \\
= a_{\omega_2}^* e^{i\omega_2 L_2} (A_k^* Q^{-1} A_{k+s})^T e^{-i\omega_2 L_2} a_{\omega_2} \\
= a_{\omega_2}^* (A_k^* Q^{-1} A_{k+s}) a_{\omega_2}. 
\]  
(6.B.7)

Making use of (6.B.7), we then rewrite (6.B.6) as

\[
\mu_s = \sum_{k=0}^{L_1 - s} (L_1 + 1 - 2k - s) a_{\omega_2}^* (A_k^* Q^{-1} A_{k+s}) a_{\omega_2}. 
\]  
(6.B.8)

Next, we make use of the following result:

**Theorem 6.3**
Let \( \Lambda = \{ \Lambda_{i,j} \} \in \mathbb{C}^{(M+1) \times (M+1)} \) and let \( a_{\omega_2} = [1 \quad e^{-i\omega_2} \quad \ldots \quad e^{-iM\omega_2}]^T \), then

\[
a_{\omega_2}^* \Lambda a_{\omega_2} = \sum_{s=-M}^{M} \psi_s e^{is\omega_2}, 
\]

where

\[
\psi_s = \sum_{k=\max(0,s)}^{\min(M+s,M)} \Lambda_{k,k-s}. 
\]

If \( \Lambda \) is Hermitian, then \( \psi_s = \overline{\psi_{-s}}. \)

**Proof:** See [LSL98].
We can thus rewrite $\mu_s$ in (6.B.8) as

$$
\mu_s = \sum_{k=0}^{L_1-s} (L_1 + 1 - 2k - s) \sum_{p=-L_2}^{L_2} \Psi(s,k,p) e^{i\omega 2p} 
$$

$$
= \sum_{p=-L_2}^{L_2} \Theta(s,p) e^{i\omega 2p} 
$$

(6.B.9)

where

$$
\Theta(s,p) = \sum_{k=0}^{L_1-s} (L_1 + 1 - 2k - s) \Psi(s,p,k) 
$$

(6.B.10)

and where

$$
\Psi(s,p,k) = \sum_{t=\max(0,p)}^{\min(L_2+p,L_2)} [A_k^* Q^{-1} A_{k+s}]_{t,l-p} 
$$

(6.B.11)

Finally, by combining (6.B.11) with (6.B.5), (6.B.9) and (6.B.10), the proof is concluded. □
6.C The Whittle-Wiggins-Robinson algorithm

The Whittle-Wiggins-Robinson algorithm (WWRA) [Whi63, WR65] is the extension to the multivariate case of the Levinson-Durbin algorithm. For more details on the WWRA algorithm, see, e.g., [SS89]. The algorithm is summarized below:

(a) Initialization:

\[
\begin{align*}
Q_0 &= S_0 = R_0 \\
R_1 &= R_1
\end{align*}
\]

(b) Order update: (for \( n = 0, \ldots, L_1 \))

\[
\begin{align*}
P_n &= R_{n+1} + A_{1}^{(n)} R_n + \ldots + A_{1}^{(n)} R_{1} \\
&= \left( \begin{array}{cccc}
I & A_{1}^{(n+1)} & \ldots & A_{n}^{(n+1)} \\
B_{n+1}^{(n+1)} & B_{n+1}^{(n+1)} & \ldots & B_{1}^{(n+1)} \\
\end{array} \right) \left( \begin{array}{cc}
I & A_{1}^{(n)} & \ldots & A_{n}^{(n)} \\
0 & B_{1}^{(n+1)} & \ldots & B_{1}^{(n+1)} \\
\end{array} \right) \\
&= \left( \begin{array}{cc}
I & -P_n S_n^{-1} \\
-P_n^* Q_n^{-1} & I \\
\end{array} \right) \\
&= \left( \begin{array}{cc}
I & A_{1}^{(n)} & \ldots & A_{n}^{(n)} \\
0 & B_{1}^{(n+1)} & \ldots & B_{1}^{(n+1)} \\
\end{array} \right) \\
Q_{n+1} &= Q_n - P_n S_n^{-1} P_n^* \\
S_{n+1} &= S_n - P_n^* Q_n^{-1} P_n
\end{align*}
\]
Chapter 7

On Efficient Implementation of the 2-D Capon Algorithm

We present an efficient implementation of the 2-D Amplitude Spectrum Capon (ASC) estimator, denoted the 2-D Burg-based ASC (2-D BASC) estimator. The implementation, which will depend only on the (forward) linear prediction matrices and the (forward) prediction error covariance matrices, can be implemented using the 2-D Fast Fourier Transform. To compute the needed prediction matrices, we make use of a recently proposed 2-D lattice algorithm, which computes the linear prediction matrices directly from the multichannel data without first computing the autocorrelation sequence.

7.1 Introduction

The problem of two-dimensional (2-D) high resolution spectral estimation has been widely studied in past literature (see, e.g., [DM84, McC82, Mar87]), as well as in more recent contributions such as [LLS98, LLL98] (see also Chapter 6). Applications occur in a wide variety of fields, such as geophysics, radio astronomy, biomedical engineering, sonar and radar, to mention a few. In many of these applications, it is of key importance to obtain computationally efficient high resolution estimates, as for example it is in synthetic aperture radar (SAR) image formation and target feature extraction [DeG98], which are becoming increasingly important in many civilian and military applications [Kle97]. Another important application is 2-D nuclear magnetic resonance (NMR) spectroscopy, where both resolution and computational complexity are of utmost importance [HKB+89]. Popular approaches include the 2-D Periodogram, and in the higher reso-
olution cases the 2-D AR spectral estimator, for which a number of efficient implementations has been suggested (see, e.g., [Mar87, ML89, TES89]).

In recent literature, there has of lately been an increased interest in non-parametric data-dependent spectral estimators, in particular the Power Spectrum Capon (PSC) and the Amplitude Spectrum Capon (ASC) spectral estimators, as well as the recently proposed Amplitude and Phase Estimation (APES) spectral estimator. In Chapter 2, it was found that the 1-D APES, as well as the 1-D PSC and the 1-D ASC methods, could be interpreted as members of the class of matched-filterbank (MAFI) spectral estimators. This result was later extended to the 2-D case in [LLS98]. With the MAFI interpretation, the APES estimator was given a simpler and more intuitive derivation, and could be seen as a Capon-like method using an estimate of the noise covariance matrix. This fact much simplifies the comparison between the estimators. Recent studies have since shown that the 1-D ASC spectral estimator will have significantly higher resolution than both the 1-D PSC and the 1-D APES spectral estimators (see Chapters 3 and 4). Similar conclusions are expected to hold in the 2-D case considered here. Furthermore, an important difference is that the ASC and the APES algorithms will retain the signal’s phase information, which the PSC estimate will not. The phase information is often needed, for instance in SAR imaging where one may use the phase to improve the final image, as well as to determine the height information in the image.

In this chapter, we present a computationally efficient implementation of the 2-D ASC spectral estimation algorithm. The algorithm is based on the recent 2-D extension of the famous Burg algorithm [Mar99] (see also Chapter 6), and is denoted the 2-D Burg-based ASC (2-D BASC). The proposed implementation, which is a 2-D extension of the 1-D BASC algorithm presented in Chapter 3, will depend only on the (forward) linear prediction matrices and can be efficiently implemented using the 2-D Fast Fourier Transform (FFT). Note that if one has no interest in the signal’s phase information, and the resolution of the 2-D PSC method is adequate, one does better by using the 2-D PSC algorithm presented in Chapter 6 as it is significantly faster than the here presented 2-D ASC algorithm.

This chapter is organized as follows: In the next section the MAFI interpretation of the filterbank spectral estimators is reviewed. In Section 7.3, the computationally efficient 2-D BASC algorithm is presented. Section 7.4 contains our numerical examples and, finally, Section 7.5 contains our conclusions.
7.2 The Matched Filterbank Approach

In the filterbank approach to spectral estimation, the spectrum is estimated by passing the measured signal through an \((L_1 + 1, L_2 + 1)\)-tap 2-D narrowband finite impulse response (FIR) filter, \(H_{\omega_1, \omega_2}\), with varying center frequencies \((\omega_1, \omega_2)\) (see, e.g., [SJL98, LLS98]). The filterlengths \(L_1\) and \(L_2\) are parameters specified by the user.

Let the \(N_1 \times N_2\) data matrix \(Z\) denote the available (stationary) 2-D data sample of which the spectrum is to be estimated. The filter output can then be written as

\[
h^*_{\omega_1, \omega_2} y_{t,s} = \alpha_{\omega_1, \omega_2} e^{i(t\omega_1 + s\omega_2)} + n_{t,s} \tag{7.2.1}
\]

for \(t = 0, \ldots, M_1, s = 0, \ldots, M_2, \omega_1, \omega_2 \in [0, 2\pi)\), where \((\cdot)^*\) and \(n_{t,s}\) denote the complex conjugate transpose and some additive colored noise term, and where

\[
M_1 = N_1 - L_1 - 1
\]
\[
M_2 = N_2 - L_2 - 1.
\]

In (7.2.1), the \((L_1 + 1)(L_2 + 1) \times 1\) filter vector, \(h_{\omega_1, \omega_2}\), is defined as

\[
h_{\omega_1, \omega_2} \triangleq \text{vec}(H_{\omega_1, \omega_2}), \tag{7.2.2}
\]

where \(\text{vec}(\cdot)\) denotes the operation of stacking the columns of a matrix on top of each other. Similarly, the \((L_1 + 1)(L_2 + 1) \times 1\) snapshot vector, \(y_{t,s}\), is defined as

\[
y_{t,s} \triangleq \text{vec}(Y_{t,s}), \tag{7.2.3}
\]

where the \((L_1 + 1) \times (L_2 + 1)\) submatrices \(Y_{t,s}\) are defined as

\[
Y_{t,s} = \begin{bmatrix}
Z_{t+L_1,s+L_2} & \cdots & Z_{t,s+L_2} \\
\vdots & \ddots & \vdots \\
Z_{t+L_1,s} & \cdots & Z_{t,s}
\end{bmatrix} \tag{7.2.4}
\]

for \(t = 0, \ldots, M_1, s = 0, \ldots, M_2\). The least-squares estimate, \(\hat{\alpha}_{\omega_1, \omega_2}\), of the complex amplitude, \(\alpha_{\omega_1, \omega_2}\), in (7.2.1) is given by

\[
\hat{\alpha}_{\omega_1, \omega_2} = h^*_{\omega_1, \omega_2} G_{\omega_1, \omega_2}, \tag{7.2.5}
\]

where

\[
G_{\omega_1, \omega_2} = \frac{1}{M_1 M_2} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} y_{k_1,k_2} e^{-i(k_1 \omega_1 + k_2 \omega_2)}. \tag{7.2.6}
\]
The problem of designing \( h_{\omega_1, \omega_2} \) as a matched filterbank (MAFI) was studied in [LLS98]. It was found that the 2-D ASC method can be interpreted as being member of the MAFI class. The corresponding filter is given by (see [LLS98] for further details):\(^1\)

\[
h_{\omega_1, \omega_2} = \frac{\hat{R}^{-1}a_L(\omega_1, \omega_2)}{a_L^*(\omega_1, \omega_2)\hat{R}^{-1}a_L(\omega_1, \omega_2)},
\]

(7.2.7)

where \( \hat{R} \) is an estimate of the sample covariance matrix. Here, \( a_L(\omega_1, \omega_2) \) is the 2-D Fourier vector, defined as

\[
a_L(\omega_1, \omega_2) \triangleq a_{L_1}(\omega_k) \otimes a_{L_2}(\omega_k)
\]

(7.2.8)

\[
a_{L_k}(\omega_k) \triangleq \begin{bmatrix} 1 & e^{-i\omega_k} & \cdots & e^{-iL_k\omega_k} \end{bmatrix}^T,
\]

(7.2.9)

where \( \otimes \) and \( (\cdot)^T \) denote the Kronecker product and the transpose, respectively. The true sample covariance matrix, \( \mathbf{R} \), is defined as

\[
\mathbf{R} \triangleq \mathbb{E}\{\mathbf{y}_{l,s}\mathbf{y}_{l,s}^*\}
\]

\[
= \begin{bmatrix}
R_0 & R_1 & \cdots & R_{L_1} \\
R_1^* & R_0 & \cdots & R_{L_1} \\
\vdots & \ddots & \ddots & \ddots \\
R_{L_1}^* & \cdots & \cdots & R_0
\end{bmatrix},
\]

(7.2.10)

where \( \mathbb{E}\{\cdot\} \) denotes the expectation, and where the block matrices \( \mathbf{R}_k \) are defined as

\[
\mathbf{R}_k \triangleq \begin{bmatrix}
r_{k,0} & r_{k,1} & \cdots & r_{k,L_2} \\
r_{k,1}^* & r_{k,0} & \ddots & \vdots \\
\vdots & \ddots & \ddots & r_{k,1} \\
r_{k,L_2}^* & \cdots & r_{k,1}^* & r_{k,0}
\end{bmatrix},
\]

(7.2.11)

where

\[
r_{k,l} \triangleq \mathbb{E}\{\mathbf{z}_{l+k,s+l}^* \mathbf{z}_{l,s}\} = r_{k,-l}^*.
\]

(7.2.12)

Note that the covariance matrix \( \mathbf{R} \) has a (Hermitian) Toeplitz-Block-Toeplitz structure. An estimate of the covariance matrix, \( \hat{\mathbf{R}} \), needed in (7.2.7), can be obtained in several different ways. We will in our numerical examples

\(^1\)Note that both the PSC and the ASC spectral estimators are constructed using the same filter. The difference lies only in how the methods estimate the amplitude spectrum (see also Section 1.2.3). Also note that both estimates will need to be scaled by the filter bandwidth to yield an estimate of the PSD (see, e.g., [LSGM86, SM97]).
study the outer-product forward-only (F) sample covariance matrix estimate, i.e.,
\[
\hat{R}^F = \frac{1}{M_1 M_2} \sum_{k=0}^{M_1} \sum_{l=0}^{M_2} y_{k,l} y_{k,l}^* ,
\]
and the forward-backward averaged (FB) sample covariance matrix estimate
\[
\hat{R}^{FB} = \frac{1}{2} \left( \hat{R}^F + J \left( \hat{R}^F \right)^T J \right)
\]
(see [JS99] for an analysis and a comparison between the use of the forward-only and the forward-backward sample covariance matrices for the 1-D Capon spectral estimator. Similar conclusions are expected to hold in the 2-D case). Here, \( J \) denotes the so-called exchange matrix, whose anti-diagonal elements are ones and all other elements are zero. It is our experience that the above outer-product covariance estimates will produce spectral estimates superior to those obtained by taking the internal structure into account (see also Chapter 6).

The 2-D ASC amplitude estimate, at frequencies \((\omega_1, \omega_2)\), is given by (7.2.5) evaluated using the filter (7.2.7), i.e.,
\[
\hat{a}_{\omega_1, \omega_2} = \frac{a^*_L (\omega_1, \omega_2) \hat{R}^{-1} G_{\omega_1, \omega_2}}{a^*_L (\omega_1, \omega_2) \hat{R}^{-1} a_L (\omega_1, \omega_2)} ,
\]
and the corresponding amplitude spectrum estimate is given as the magnitude square of (7.2.15), i.e.,
\[
\hat{\phi}_{\omega_1, \omega_2} = |\hat{a}_{\omega_1, \omega_2}|^2 .
\]
Note that the 2-D ASC will in general yield a different and often preferable spectral estimate, than the 2-D PSC spectral estimator, which is obtained by estimating the power of the filter output, i.e.,
\[
\hat{\phi}^{PSC} = h^*_L (\omega_1, \omega_2) \hat{R} h_{\omega_1, \omega_2} = \frac{1}{a^*_L (\omega_1, \omega_2) \hat{R}^{-1} a_L (\omega_1, \omega_2)}
\]
(see also the discussion in Section 1.2.3). The problem of interest in this chapter is to compute (7.2.15) in a computationally efficient manner. An efficient implementation of the 2-D PSC spectral estimate in (7.2.17) was proposed in Chapter 6. We note that the primary computational burden to evaluate (7.2.15) is not, as it might first seem, that associated with the inverse of the large dimension matrix \( \hat{R} \). If that was the case, an efficient algorithm for the inversion of a Toeplitz-Block-Toeplitz matrix could have been used [WK83, KCM84]. Rather, it is the computation of (7.2.15) over all frequencies that is normally more time consuming. In the following section, we propose an efficient way to do this using the 2-D Fast Fourier Transform (FFT).
7.3 Proposed Efficient Implementation

As was suggested in [SJL98, LLL98], let $\mathbf{C} \triangleq \mathbf{R}^{-1/2}$ denote a square root of the positive definite matrix $\mathbf{R}^{-1}$ defined in (7.2.10), and let

$$\nu_{\omega_1, \omega_2} \triangleq \mathbf{C}^* \mathbf{a}_L(\omega_1, \omega_2)$$

(7.3.18)

$$\mu_{\omega_1, \omega_2} \triangleq \mathbf{C}^* \mathbf{G}_{\omega_1, \omega_2} = \frac{\mathbf{C}^* \mathbf{W} \mathbf{a}_M(\omega_1, \omega_2)}{M_1 M_2}$$

(7.3.19)

where

$$\mathbf{W} = [\mathbf{y}_{0,0} \cdots \mathbf{y}_{M_1,0} \cdots \mathbf{y}_{0,M_2} \cdots \mathbf{y}_{M_1,M_2}].$$

(7.3.20)

Making use of (7.2.17), as well as (7.3.18), the 2-D PSC spectral estimate can be formulated as

$$\varphi_{PSC}^{\omega_1, \omega_2} = \frac{1}{\nu_{\omega_1, \omega_2}^* \nu_{\omega_1, \omega_2}}.$$  

(7.3.21)

Similarly, the 2-D ASC spectral estimate in (7.2.16) can be found as

$$\varphi_{ASC}^{\omega_1, \omega_2} = \left| \frac{\nu_{\omega_1, \omega_2}^* \mu_{\omega_1, \omega_2}}{\nu_{\omega_1, \omega_2}^* \nu_{\omega_1, \omega_2}} \right|^2.$$  

(7.3.22)

Thus, both the 2-D PSC and the 2-D ASC spectral estimators can, given an estimate of $\mathbf{C}$, both be efficiently computed using the 2-D FFT.

An efficient computation of (7.3.22) was recently proposed in [LLL98]. There, $\mathbf{C}$ was estimated by computing the Cholesky factorization of the inverted outer-product sample covariance estimate $\hat{\mathbf{R}}$. This approach requires the computing of $\hat{\mathbf{R}}$, its inverse as well as the Cholesky factor $\hat{\mathbf{C}}$. Here, we instead propose to construct $\mathbf{C}$ from the (forward) linear prediction matrices (see, e.g., [SS89], Complement C8.3)

$$\hat{\mathbf{C}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{1,L_1}^* & \mathbf{I} \\ \vdots & \ddots & \ddots \\ \mathbf{A}_{L_1,L_1}^* & \cdots & \mathbf{A}_{1,1}^* & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{L_1}^{-1/2} & \mathbf{0} \\ \mathbf{U}_{L_1-1}^{-1/2} & \ddots \\ \mathbf{0} & \cdots & \mathbf{U}_0^{-1/2} \end{bmatrix}$$

(7.3.23)

where $\{\mathbf{A}_{k,n}\}$ and $\mathbf{U}_n$ denote the matrix coefficients and prediction error covariance matrices of the (forward) linear prediction model of order $n$ (see [Mar87, SS89] for further details). Here, $\mathbf{U}_0 = \mathbf{R}_0$. To compute the needed matrix square roots in (7.3.23), we use the Cholesky factorization.
As these matrices are significantly smaller than the full covariance matrix \( \hat{R} \), the computational burden of doing so will in comparison be minor.

In Chapter 6, it was found that the efficient implementation of the 2-D PSC spectral estimate computed using linear prediction matrices as obtained by the novel 2-D lattice algorithm, will be noticeably better than the spectral estimates obtained by computing the linear prediction matrices using the Whittle-Wiggins-Robinson algorithm (WWRA) [Whi63, WR65] which is based on an estimate of the 2-D covariance matrix. Similarly, it is our experience that we will obtain better spectral estimates in our proposed implementation if we use the linear prediction matrices as obtained by the 2-D lattice algorithm, as compared to the ones obtained from the WWRA. This is well in accordance with similar results in the 1-D case [UB75].

Note that the 2-D BASC estimator will produce (almost) the same estimate as the 2-D ASC estimator computed from the forward-backward averaged covariance matrix estimate in (7.2.14). This forward-backward ASC (FB-ASC) will yield a significantly better spectral estimate, although with a somewhat lower resolution, than the forward-only ASC (F-ASC) estimator [LLS98]. The reason that BASC will produce only almost the same estimate as FB-ASC can be explained as follows: the estimate of \( C \), as obtained by using either the WWRA or the 2-D lattice algorithm, will in general not yield the same estimate of \( C \) as the Cholesky factorization of the covariance matrix estimate in (7.2.14) will. This is due to the fact that \( \hat{R} \) is the outer product covariance matrix estimate, and will thus not have the Toeplitz-Block-Toeplitz structure that is obtained if an estimate of \( R \) is constructed from a \( C \) computed by the WWRA or the 2-D lattice algorithm. Thus, the BASC and the FB-ASC spectral estimates will not be identical, although as is shown in the next section the estimates are basically the same.

### 7.4 Numerical Examples

We first study the computational complexity of the different implementations. In our first example, the data matrix has been generated as the sum of four 2-D complex sinusoids (cisoids) corrupted by additive complex Gaussian white noise. The computational complexity is evaluated as the data matrix size, \( N_1 = N_2 \), varies. Figure 7.1 illustrates the relative computational complexity of the proposed 2-D BASC spectral estimator for varying data matrix dimensions, as compared to the 2-D PSC and 2-D ASC methods proposed in [LLL98] as well as the 2-D PSC estimator in Chapter 6. In the figure, the computational load of the different estimators, as measured by MATLAB, has been normalized with the 2-D ASC method proposed...
in [LLL98]. The simulation shows that the proposed 2-D BASC estimator will be clearly faster than the 2-D ASC estimator in [LLL98], especially for larger matrices. As the difference between the two estimators basically lies in how the $(L_1 + 1) \times (L_2 + 1)$ Cholesky matrix, $C$, is computed, this comes as no surprise. From the figure, it can also be seen that the 2-D PSC implementation in Chapter 6 will be about five times faster than the 2-D PSC implementation in [LLL98]. The reader is reminded that the 2-D BASC algorithm should only be used in cases where one wishes to retain the signal’s phase information, or when the resolution of the 2-D PSC algorithm is not sufficient. In the example, the filterlengths were $L_1 = L_2 = N_1/4$, and the spectrum is zeropadded to length $N_{\omega_1} = N_{\omega_2} = 4N_1$ (which means that the spectrum is evaluated using a $4N_1 \times 4N_1$-point 2-D FFT).

![Relative computational complexity vs data matrix size](image)

**Figure 7.1:** Relative computational complexity vs data matrix size ($N_1 = N_2$).

We proceed by illustrating the spectral resolutions achieved by the different methods. First, we use a $32 \times 32$ data matrix that consists of a sum of four cisoids corrupted by additive complex Gaussian white noise. The cisoids all have unit amplitude, a phase offset of $\pi/4$, and are located at $f = (-0.25, 0.25), (0, 0), (0.03, 0), (0.3, -0.1)$. Figures 7.2(a)–(c) illustrate the resulting resolutions obtained by using the 2-D PSC and 2-D ASC estimates of [LLL98], as well as the 2-D PSC estimate presented in Chapter 6. The proposed 2-D BASC estimate is shown in Figure 7.2(d). As seen from the figure, the ASC estimators will have better resolution than the PSC estimators. In the example, the filter lengths were $L_1 = L_2 = N_1/4$, and the spectrum is zeropadded to length $N_{\omega_1} = N_{\omega_2} = 4N_1$. 

![Data matrix size](image)

![Relative computational load](image)
Figure 7.2: Illustration of the resolution and accuracy for the different spectral estimators. The estimates are plotted for fractions of the sampling frequency. 
(a) The 2-D PSC estimate of [LLL98]. (b) The 2-D ASC estimate of [LLL98], (c) The 2-D PSC estimate presented in Chapter 6. (d) The proposed 2-D BASC estimate.
Figure 7.3: Illustration of the resolution and accuracy for the different spectral estimators. (a) The 2-D PSC estimate of [LLL98], (b) The 2-D ASC estimate of [LLL98], (c) The 2-D PSC estimate presented in Chapter 6. (d) The proposed 2-D BASC estimate.
Finally, we consider synthetic aperture radar (SAR) imaging of a simulated MIG-25 airplane. The $32 \times 32$ data matrix was provided by the Naval Research Laboratory. Figures 7.3(a)–(d) show the $128 \times 128$ SAR images as obtained by the different methods, using $L_1 = L_2 = 15$.

7.5 Conclusions

In this work we have presented an efficient implementation of the 2-D ASC spectral estimator, denoted the 2-D BASC estimator. The derivation is based on the (forward) linear prediction matrices and the (forward) prediction error covariance matrices, and can be implemented using the 2-D FFT. It is found that the presented implementation will yield significant computational gains as compared to the recently proposed approach by Liu et al [LLL98], especially for larger matrix sizes.
Part II

Model-Based Signal Analysis
Chapter 8

Cisoid Parameter Estimation in the Colored Noise Case

The problem of estimating the parameters of complex-valued sinusoidal signals (cisoids, for short) from data corrupted by colored noise occurs in many signal processing applications. We present a simple formula for the asymptotic (large-sample) Cramér-Rao bound (CRB) matrix associated with this problem. The maximum likelihood method (MLM), which estimates both the signal and noise parameters, attains the performance corresponding to the asymptotic CRB, as the sample length increases. More interestingly, we show that a computationally much simpler nonlinear least-squares method (NLSM), which estimates the signal parameters only, achieves the same performance in large samples.

8.1 Introduction and Preliminaries

Consider the following noisy observations of a complex-valued discrete-time sinusoidal signal:

\[ y(t) = x(t, \theta) + \varepsilon(t), \quad t = 0, \ldots, N - 1 \]  

(8.1.1)

where \( N \) is the number of available observations, \( \varepsilon(t) \) is an interference and measurement noise, and

\[ x(t, \theta) = \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \varphi_k)} \]  

(8.1.2)

\[ \theta = \left[ \alpha_1 \varphi_1 \omega_1 \cdots \alpha_n \varphi_n \omega_n \right]^T \]  

(8.1.3)

with \( [\cdot]^T \) denoting transposition.
We assume that:

**A1.** The components of $\theta$ are unknown constants that satisfy:

$$\alpha_k > 0, \quad \varphi_k \in [0, 2\pi], \quad \omega_k \in [0, 2\pi], \quad \omega_k \neq \omega_j \quad (\text{for } k \neq j)$$

**A2.** The noise $\varepsilon(t)$ is a zero mean stationary process with a positive and piecewise continuous spectral density $\phi(\omega)$; The possible discontinuities of $\phi(\omega)$ do not appear at any $\omega = \omega_k$ for $k = 1, \ldots, n$.

**A3.** The noise spectral density $\phi(\omega)$ is parameterized by a finitely dimensional unknown vector $\mu$ that does not have any component in common with $\theta$. Furthermore, the noise process $\{\varepsilon(t)\}$ is circularly symmetric Gaussian distributed.

Assumption **A1** does not introduce any restriction.

Assumption **A2** is also quite general. In particular, let $\varepsilon(t)$ be a member of the large class of linear processes. This means that $\varepsilon(t)$ can be written as:

$$\varepsilon(t) = \sum_{k=-\infty}^{\infty} h_k \xi(t-k) \quad (8.1.4)$$

where $\{\xi(t)\}$ is white noise with zero mean and unit variance. Then, under weak conditions on $\{h_k\}$, the spectral density of $\varepsilon(t)$

$$\phi(\omega) \triangleq \sum_{k=-\infty}^{\infty} E[\varepsilon(t)\varepsilon^*(t-k)]e^{-i\omega k} = \left| \sum_{k=-\infty}^{\infty} h_k e^{-i\omega k} \right|^2 \quad (8.1.5)$$

satisfies **A2** (see, e.g., [GR56]). Note that in the complex-valued case under discussion $\phi(\omega)$ is generally different from $\phi(-\omega)$. Hence the definition in (8.1.5) of $\phi(\omega)$ must be observed when evaluating the various expressions, which depend on $\phi(\omega)$, that appear in the following sections.

Assumption **A3** evidently does introduce some restrictions. For instance, the condition in **A3** that $\phi(\omega)$ is a known function of a finitely-dimensional unknown vector is satisfied by ARMA (autoregressive moving average) noises but not necessarily by the general linear noise process in equation (8.1.4). The aforementioned condition is essentially required by the CRB and MLM analysis in the next section. The Gaussian hypothesis in **A3**, on the other hand, can be relaxed without affecting the asymptotic distributional properties of the MLM. However, without the Gaussian assumption the above-mentioned method can no longer be interpreted as a MLM, nor can its covariance matrix anymore be interpreted as the CRB. Regarding
the NLSM, as it will become clear shortly, this method does not depend on A3 at all, which of course is an added bonus (as we have already stated the NLSM is also much simpler computationally than the MLM, see the next sections for details).

The estimation of the parameter vector $\theta$ in (8.1.1) is a problem that occurs in a large number of signal processing applications (see, e.g., [Kay88, Mar87] and the references therein). In the case of white noise scenarios there is a wealth of good methods that can be used to solve this problem (see the cited references once again). The more difficult but definitely more practically relevant case of colored noise has only recently begun to receive the necessary attention. See, for example, [FF95, KN94, CKB87, TM94, SN89]. Of the difficulties associated with the latter case, the following are worth mentioning:

a) The MLM estimation of the parameters in the noisy signal model (8.1.1) (both $\theta$ and $\mu$) leads to an involved computational problem. Approximate ML methods have been proposed to reduce the computational burden associated with the exact MLM (see, e.g., [KN94, CKB87, TM94]). However, the latter methods still estimate both the signal and noise parameters and hence the simplifications which they achieve are not significant. The use of non-ML methods to achieve computational simplicity is possible (for example, extended Yule-Walker methods can easily be derived for moving average (MA) noise scenarios), but is not advisable owing to the poor statistical accuracy that may be associated with such methods.

b) The exact CRB matrix corresponding to the signal model in (8.1.1)–(8.1.3) is readily derived [FF95]. However, it has an unenlightening expression whose evaluation may be computationally burdensome (especially in large samples).

In this paper we aim at providing solutions to the problems described above. Regarding (a), we show that the ultimate large-sample statistical performance of the exact MLM estimator of the signal parameters in $\{\theta\}$ is also achievable by a computationally much simpler NLSM which ignores (and hence does not estimate) the noise correlation. Note that the noise parameters are usually nuisances for the estimation problem under discussion, and hence their estimation is not required in most cases. Concerning (b), we derive an asymptotic (large-sample) CRB matrix for the parameter vector $\theta$, which has a simple and easily interpretable expression.

In closing this section we note that the case of real-valued sinusoidal signals-in-colored-noise has been treated in [SN89] to which we refer the reader for the real counterparts of the complex CRB, MLM and NLSM
discussed herein. In the cited paper the asymptotic CRB matrix was derived as the large-sample covariance matrix of the MLM. Herein we take the other possible route: we obtain the asymptotic CRB, and hence the covariance matrix of the large-sample distribution of the MLM, as the limit of the finite-sample CRB matrix when \( N \) tends to infinity. Compared with the analysis in [SN89], the present analysis is based on some basic results in [GR56] and is considerably simpler.

## 8.2 MLM and CRB

Let

\[
\mathbf{y} = \begin{bmatrix} y(0) & \cdots & y(N-1) \end{bmatrix}^T \\
\mathbf{x}(\boldsymbol{\theta}) = \begin{bmatrix} x(0, \theta) & \cdots & x(N-1, \theta) \end{bmatrix}^T \\
\mathbf{\varepsilon} = \begin{bmatrix} \varepsilon(0) & \cdots & \varepsilon(N-1) \end{bmatrix}^T
\]

and let \( Q(\mu) \) denote the parameterized form of the covariance matrix of the noise vector \( \mathbf{\varepsilon} \) (\( Q(\mu) \) is Hermitian and Toeplitz). With this notation, and under the assumptions made, the negative log-likelihood function associated with the observed data vector \( \mathbf{y} \) is given by (to within an additive constant):

\[
f_{\text{ML}}(\mathbf{\eta}) = \ln |Q(\mu)| + (\mathbf{y} - \mathbf{x}(\boldsymbol{\theta}))^* Q^{-1}(\mu) (\mathbf{y} - \mathbf{x}(\boldsymbol{\theta})) \tag{8.2.6}
\]

where the superscript * denotes the conjugate transpose, and

\[
\mathbf{\eta} = \begin{bmatrix} \mu^T & \theta^T \end{bmatrix}^T. \tag{8.2.7}
\]

The inverse matrix \( [Q(\mu)]^{-1} \) in (8.2.6) is guaranteed to exist for all values of \( N \) by the assumption that \( \phi(\omega, \mu) > 0 \) for any \( \omega \in [0, 2\pi) \) (see A2).

The exact ML estimates of \( \theta \) and \( \mu \) are obtained by the minimization of the function in (8.2.6). Under the assumptions A1-A3 the asymptotic (large sample) distribution of the MLM estimate is Gaussian with mean equal to the true parameter vector and covariance matrix equal to the asymptotic CRB (see, e.g., [GR56] and the references therein). In the remaining part of this section we derive an expression for the asymptotic CRB matrix. A reason for being interested in the asymptotic expression for the CRB, rather than in its exact finite-sample expression, is that only the former is achievable by the MLM and possibly by other methods as well. (In the estimation problem under discussion the finite-sample CRB performance is not attained by any known methods, and in fact it may not
be achievable at all.) Another reason for being interested in an asymptotic CRB formula lies in the simplicity of such a formula, as opposed to the complexity and lack of insights corresponding to the exact CRB expression (see [SMFS89] for more details on this aspect).

It is well known that the (exact) CRB matrix for the problem under study is given elementwise by (see, e.g., [Ban71, SN90, SM97]):

\[
[\text{CRB}^{-1}(\hat{\eta})]_{kp} = \text{tr}\left[ Q^{-1}(\mu) \frac{\partial Q(\mu)}{\partial \eta_k} Q^{-1}(\mu) \frac{\partial Q(\mu)}{\partial \eta_p} \right] \\
+ 2 \Re \left[ \frac{\partial^* x(\theta)}{\partial \eta_k} Q^{-1}(\mu) \frac{\partial x(\theta)}{\partial \eta_p} \right] 
\]

where the right-hand side should be evaluated at the true value of the parameter vector \( \eta \). Henceforth \( \text{tr}(\cdot) \) denotes the trace operator, and \( \Re(\cdot) \) denotes the real part of the quantity between parentheses. Also, the notation \( \text{CRB}(\hat{\eta}) \) is used to mean that the CRB under discussion is associated with estimates \( \hat{\eta} \) of the parameter vector \( \eta \). By assumption, the parameter subvectors \( \mu \) and \( \theta \) do not share any elements. By using this fact one can readily verify that the \( \text{CRB}(\hat{\eta}) \) matrix is block diagonal (see, e.g., [FF95]):

\[
\text{CRB}(\hat{\eta}) = \begin{bmatrix} \text{CRB}(\hat{\mu}) & 0 \\ 0 & \text{CRB}(\hat{\theta}) \end{bmatrix}
\]

where the block corresponding to the signal parameter vector is given by:

\[
[\text{CRB}(\hat{\theta})]^{-1} = 2\Re \left[ \frac{\partial^* x(\theta)}{\partial \theta} Q^{-1}(\mu) \frac{\partial x(\theta)}{\partial \theta^T} \right].
\]

The expression for the CRB block corresponding to the noise parameters is also easily obtained from (8.2.8) but, as already stated, usually we do not have any interest in those parameters.

The following additional notation is required to proceed with the derivation of the \( \text{CRB}(\hat{\theta}) \)

\[
a_k = \begin{bmatrix} 1 & e^{i\omega_k} & \cdots & e^{i(N-1)\omega_k} \end{bmatrix}^T e^{i\varphi_k} \\
d_k = i \begin{bmatrix} 0 & e^{i\omega_k} & \cdots & (N-1)e^{i(N-1)\omega_k} \end{bmatrix}^T e^{i\varphi_k}
\]

for \( k = 1, \ldots, n \). By using this notation we can write \( x(\theta) \) as:

\[
x(\theta) = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}
\]

\(^1\)Equation (8.2.8) is sometimes called the extended Bangs-Slepian formula.
from which it readily follows that
\[
\frac{\partial \mathbf{x}(\theta)}{\partial \theta^T} = \begin{bmatrix}
\mathbf{a}_1 & i\alpha_1 \mathbf{a}_1 & \alpha_1 \mathbf{d}_1 & \cdots & \mathbf{a}_n & i\alpha_n \mathbf{a}_n & \alpha_n \mathbf{d}_n
\end{bmatrix}.
\] (8.2.14)

Insertion of (8.2.14) into (8.2.10) yields a formula for the exact (finite-sample) \( \text{CRB}(\hat{\theta}) \) (see, e.g., [FF95] for more details on the exact CRB). As already stated, the exact \( \text{CRB}(\hat{\theta}) \) formula is neither insightful nor easily interpretable. Additionally its evaluation can be computationally burdensome especially in cases with relatively large values of \( N \).

Let the \((3n \times 3n)\) matrix
\[
\mathbf{K}_N = \begin{bmatrix}
\begin{array}{ccc}
N^{1/2} & 0 & 0 \\
0 & N^{1/2} & 0 \\
n & 0 & N^{1/2}
\end{array}
\end{bmatrix}
\] (8.2.15)

In what follows we make use of results in [GR56], Chapter 7, to show that the following asymptotic normalized CRB matrix has a simple expression:

\[
\text{As CRB}(\hat{\theta}) = \lim_{N \to \infty} \mathbf{K}_N \text{ CRB}(\hat{\theta}) \mathbf{K}_N.
\] (8.2.16)

Before being able to do so, however, we need some preparations. Let
\[
\mathbf{g}(t) \triangleq \frac{\partial \mathbf{x}(t, \theta)}{\partial \theta} = \begin{bmatrix}
e^{i(\omega_1 t + \varphi_1)} \\
i\alpha_1 e^{i(\omega_1 t + \varphi_1)} \\
i\alpha_1 e^{i(\omega_1 t + \varphi_1)} \\
\vdots \\
e^{i(\omega_n t + \varphi_n)} \\
i\alpha_n e^{i(\omega_n t + \varphi_n)} \\
i\alpha_n e^{i(\omega_n t + \varphi_n)}
\end{bmatrix}
\] (8.2.17)

and let
\[
\mathbf{G} \triangleq \frac{\partial \mathbf{x}(\theta)}{\partial \theta^T} = \begin{bmatrix}
g^T(0) \\
\vdots \\
g^T(N - 1)
\end{bmatrix}.
\] (8.2.18)

By making use of the previous notation we can rewrite (8.2.10) as follows
\[
\text{CRB}(\theta) = \frac{1}{2} [\Re \{\mathbf{G}^* \mathbf{Q}^{-1} \mathbf{G}\}]^{-1}.
\] (8.2.19)
Following [GR56], let us introduce the following “covariance sequence”,

\[
R_\tau \triangleq \lim_{N \to \infty} K_N^{-1} \left[ \sum_{t=0}^{N-1} g(t + \tau) g^*(t) \right] K_N^{-1}
\]  

(8.2.20)

which, by a relatively simple algebra, can equivalently be written as

\[
R_\tau = \left[ \begin{array}{cc}
  e^{i\tau \omega_1} I_3 & 0 \\
  0 & e^{i\tau \omega_n} I_3
\end{array} \right] 
\times \lim_{N \to \infty} \left[ K_N^{-1} (G^* G)^T K_N^{-1} \right],
\]

(8.2.21)

where \( I_k \) denote the identity matrix of size \( k \).

Hereafter the superscript "+" denotes the complex conjugate. A straightforward but somewhat tedious calculation shows that the generic \((k,p)\) block of the limit matrix in (8.2.21) is given by:

\[
\left\{ \lim_{N \to \infty} \left[ \begin{array}{ccc}
  N^{-\frac{1}{2}}a_k^* & N^{-\frac{1}{2}}a_p & 0 \\
  -N^{-\frac{1}{2}}i\alpha_k a_k^* & N^{-\frac{1}{2}}i\alpha_p a_p & 0 \\
  N^{-\frac{1}{2}}\alpha_k d_k^* & N^{-\frac{1}{2}}\alpha_p d_p & 0 \\
\end{array} \right] \right\}^+
\]

\[
= \left[ \begin{array}{ccc}
  1 & \frac{\alpha_k}{\alpha_p} & \frac{\alpha_k}{\alpha_p} \\
  -\frac{\alpha_k}{\alpha_p} & \frac{\alpha_k^2}{\alpha_p^2} & \frac{\alpha_k^2}{\alpha_p^2} \\
  -\frac{i\alpha_k}{\alpha_p} & \frac{\alpha_k^2}{\alpha_p^2} & \frac{\alpha_k^2}{\alpha_p^2} \\
\end{array} \right]^+
\]

\[
\delta_{k,p} \triangleq B_k \delta_{k,p}
\]

(8.2.22)

where \( \delta_{k,p} \) is the Kronecker delta. By combining (8.2.21) and (8.2.22) we obtain:

\[
R_\tau = \left[ \begin{array}{cc}
  B_1 e^{i\tau \omega_1} & 0 \\
  0 & \cdots \\
  0 & B_n e^{i\tau \omega_n}
\end{array} \right].
\]

(8.2.23)

The matrices \( \{B_k\} \) are nonnegative definite by construction. It follows from this observation and (8.2.23) that the sequence \( \{R_\tau\} \) admits the spectral representation:

\[
R_\tau = \int_{-\pi}^{\pi} \Psi(\omega) e^{i\tau \omega} d\omega
\]

(8.2.24)

where

\[
\Psi(\omega) = \left[ \begin{array}{ccc}
  B_1 \delta(\omega - \omega_1) & 0 \\
  0 & \cdots \\
  0 & B_n \delta(\omega - \omega_n)
\end{array} \right]
\]

(8.2.25)
and where $\delta(\omega - \omega_k)$ denotes the Dirac impulse located at $\omega = \omega_k$.\footnote{The spectral representation in (8.2.24) could have been written more formally as a Fourier-Stieltjes integral (see, e.g., [GR56]).} Observe that $\Psi(\omega)$ is a nonnegative definite matrix for any $\omega \in [0, 2\pi]$, as it should.

We are now in a position to make use of a result from [GR56] to evaluate the limit in (8.2.16). Specifically it follows from [GR56], Chapter 7, and the above calculations leading to (8.2.24), (8.2.25) that:

\[
\text{As CRB} (\hat{\theta}) = \frac{1}{2} \left\{ \Re \left[ \lim_{N \to \infty} K_N^{-1} G^* Q^{-1} G K_N^{-1} \right] \right\}^{-1}
\]

\[
= \frac{1}{2} \left\{ \Re \left[ \int_{-\pi}^{\pi} \frac{1}{\phi(\omega)} \Psi(\omega) d\omega \right] \right\}^{-1}
\]

\[
= \frac{1}{2} \left[ \begin{array}{ccc}
\phi(\omega_1) [\Re(B_1)]^{-1} & 0 & \cdots \\
0 & \phi(\omega_n) [\Re(B_n)]^{-1} & \cdots \\
0 & 0 & \phi(\omega_n) [\Re(B_n)]^{-1}
\end{array} \right] \quad (8.2.26)
\]

where the last equality was obtained by using (8.2.25). As

\[
[\Re(B_k)]^{-1} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \frac{\sigma_k^2}{\alpha_k^2} & \frac{\sigma_k^2}{3} \\
0 & \frac{\sigma_k^2}{3} & \frac{\sigma_k^2}{3}
\end{bmatrix}^{-1} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \frac{\alpha_k^2}{\sigma_k^2} & \frac{\alpha_k^2}{6} \\
0 & \frac{\alpha_k^2}{\sigma_k^2} & \frac{\alpha_k^2}{12}
\end{bmatrix} \quad (8.2.27)
\]

it readily follows that

\[
\text{As CRB} (\hat{\theta}) = \frac{1}{2} \begin{bmatrix}
\mathbf{A} \mathbf{C}_1 & 0 \\
0 & \ddots \\
0 & 0 & \mathbf{A} \mathbf{C}_n
\end{bmatrix} \quad (8.2.28)
\]

where

\[
\mathbf{A} \mathbf{C}_k = \begin{bmatrix}
\phi(\omega_k) & 0 & 0 \\
0 & \frac{4\phi(\omega_k)}{\alpha_k^2} & -\frac{6\phi(\omega_k)}{\alpha_k^2} \\
0 & \frac{6\phi(\omega_k)}{\alpha_k^2} & \frac{12\phi(\omega_k)}{\alpha_k^2}
\end{bmatrix}.
\]

The conceptual and computational simplicity of the As CRB ($\hat{\theta}$) matrix are now apparent. Note that the ratio $\phi(\omega_k)/\alpha_k^2$ in (8.2.28) can be interpreted as being the inverse of the local signal-to-noise ratio (SNR) for the $k$th cisoid.

The formula in (8.2.28) for the asymptotic CRB matrix has since being formulated been found useful in a number of studies, e.g., a study of the effects of sampling and quantization on single-tone frequency estimation.
[HMH99], and a study on parameter estimation of harmonic sinusoidal signals [LSL].

To conclude this section we summarize the previously derived results, for the reader’s convenience:

(i) The exact CRB$(\hat{\eta})$ matrix is block diagonal with the block corresponding to the signal parameters given by (8.2.10), (8.2.14). The aforementioned CRB formula is not easily interpretable, nor may it be easy to compute.

(ii) As the sample length $N$ increases the normalized CRB matrix, i.e., $K_N$ CRB$(\hat{\theta})$ $K_N^{-1}$, tends to the As CRB$(\hat{\theta})$ given in (8.2.28). The derived formula for the asymptotic CRB matrix is both easy to interpret and simple to compute.

(iii) The MLM estimator asymptotically achieves the performance corresponding to the As CRB. More exactly, the covariance matrix of the normalized ML estimation errors $K_N(\hat{\theta} - \theta)$ is asymptotically given by (8.2.28). The exact MLM, however, is typically difficult to obtain owing to the complexity of the optimization problem associated with it.

8.3 NLSM

The NLSM estimates the signal parameter vector $\theta$ by minimizing the function in (8.2.6) with $Q(\mu)$ set to the identity matrix:

$$f_{NLS}(\theta) = \frac{1}{2} [y - x(\theta)]^*[y - x(\theta)],$$  \hspace{1cm} (8.3.29)

where the factor $\frac{1}{2}$ has been introduced for the sake of convenience. Hence, if the noise was white then the NLSM would coincide with the MLM. Interestingly enough, the NLSM asymptotically achieves the same statistical performance as the MLM even in the colored noise case. To prove the previous assertion we first make use of a standard Taylor series expansion technique to obtain the following asymptotically valid expression for the normalized errors in the NLS estimate:

$$K_N(\hat{\theta} - \theta) = \{WK_N^{-1}\}^{-1} \left\{K_N^{-1}R \left[\frac{\partial x^*(\theta)}{\partial \theta} \varepsilon \right] \right\},$$  \hspace{1cm} (8.3.30)

where

$$W = K_N^{-1}R \left[\frac{\partial x^*(\theta)}{\partial \theta} \frac{\partial x(\theta)}{\partial \theta^T} - \sum_{t=0}^{N-1} \frac{\partial^2 x^*(t, \theta)}{\partial \theta^2} \varepsilon(t) \right].$$
By a straightforward but tedious calculation, which is omitted in the interest of brevity, it is possible to show that the term

\[ \mathbf{K}_n^{-1} \sum_{t=0}^{N-1} \frac{\partial^2 x^*(t, \theta)}{\partial \theta^2} \varepsilon(t) \mathbf{K}_n^{-1} \]

in (8.3.30) is asymptotically negligible (we refer to [SN89] for a similar calculation in the real-valued case). Combining this observation with (8.3.30) yields the following asymptotic expression for the normalized covariance matrix of the NLSM:

\[
\text{As Cov} (\theta) \triangleq \lim_{N \to \infty} E \left[ \mathbf{K}_N(\theta - \theta) \left[ \mathbf{K}_N(\theta - \theta) \right]^T \right] = \lim_{N \to \infty} \mathbf{K}_N \left[ \mathcal{R}(\mathbf{G}^* \mathbf{G}) \right]^{-1} \Lambda \times \left[ \mathcal{R}(\mathbf{G}^* \mathbf{G}) \right]^{-1} \mathbf{K}_N
\]

where

\[
\Lambda = E \left\{ \left[ \left[ \mathcal{R}(\mathbf{G}^* \varepsilon) \right] \left[ \mathcal{R}(\mathbf{G}^* \varepsilon)^T \right] \right] \right\}
\]

By a simple algebra we obtain,

\[
\Lambda = \frac{1}{2} \mathcal{R} \left[ E(\mathbf{G}^* \varepsilon \varepsilon^* \mathbf{G}) \right] + \frac{1}{2} \mathcal{R} \left[ E(\mathbf{G}^* \varepsilon \varepsilon^T \mathbf{G}^+) \right] = \frac{1}{2} \mathcal{R}(\mathbf{G}^* \mathbf{Q} \mathbf{G})
\]

where the last equality follows from the assumption that \( \varepsilon \) is circularly symmetric distributed. Insertion of (8.3.32) into (8.3.31) yields:

\[
\text{As Cov} (\theta) = \frac{1}{2} \left[ \mathbf{K}_n^{-1} \mathcal{R}(\mathbf{G}^* \mathbf{G}) \mathbf{K}_n^{-1} \right]^{-1} \times \mathcal{R} \left( \mathbf{K}_n^{-1} \mathbf{G}^* \mathbf{Q} \mathbf{G} \mathbf{K}_n^{-1} \right) \times \left[ \mathbf{K}_n^{-1} \mathcal{R}(\mathbf{G}^* \mathbf{G}) \mathbf{K}_n^{-1} \right]^{-1}.
\]

Clearly, (cf. (8.2.21) and (8.2.23))

\[
\lim_{N \to \infty} \mathcal{R}(\mathbf{K}_n^{-1} \mathbf{G}^* \mathbf{G} \mathbf{K}_n^{-1}) = \mathcal{R}(\mathbf{R}_0)
\]

\[
= \begin{bmatrix}
\mathcal{R}(\mathbf{B}_1) & 0 & \cdots \\
0 & \ddots & \ddots \\
0 & \cdots & \mathcal{R}(\mathbf{B}_n)
\end{bmatrix}.
\]

Furthermore, it follows from a result in [GR56], Chapter 7, similar to the one used to derive (8.2.26), that:

\[
\lim_{N \to \infty} \mathcal{R}(\mathbf{K}_n^{-1} \mathbf{G}^* \mathbf{Q} \mathbf{G} \mathbf{K}_n^{-1}) = \mathcal{R} \left[ \int_{-\pi}^{\pi} \phi(\omega) \Psi(\omega) d\omega \right]
\]
\[
\begin{bmatrix}
\phi(\omega_1) \mathbb{R}(B_1) & 0 \\
0 & \ddots \\
0 & \phi(\omega_n) \mathbb{R}(B_n)
\end{bmatrix}
\]

(8.3.35)

where the last equality has been obtained by using (8.2.25). Combining (8.3.33)–(8.3.35) leads to an expression for \( \text{As Cov}(\hat{\theta}) \) which coincides with (8.2.26). Thus, we have proved the following equality:

\[
\text{As Cov}(\hat{\theta}) = \text{As CRB}(\hat{\theta})
\]

(8.3.36)

which is exactly what was asserted before.

As the nuisance parameters characterizing the noise do not appear in (8.3.29), the NLSM estimates are significantly easier to obtain than the MLM estimates. A computationally efficient and reliable relaxation-based (RELAX) algorithm that can be used to determine the NLSM estimates has been introduced in [LS96b] (see also [LL98] for an efficient implementation of the RELAX algorithm).

### 8.4 Numerical Examples

From a practical standpoint the most important result of the present paper is apparently the fact that the NLSM asymptotically achieves the ultimate statistical performance corresponding to the As CRB(\( \hat{\theta} \)). In this section we provide empirical evidence which suggests that the aforementioned asymptotic result can hold in samples with modest practical lengths. We compute the NLS estimates by using the RELAX algorithm of [LS96b], and the As CRB(\( \hat{\theta} \)) by using (8.2.28). As there is apparently no competitive algorithm for computing the ML estimates we do not consider these estimates in the following simulation examples (the approximate MLM algorithms in [KN94] and [CKB87], for instance, appear to require considerable computer times at a statistical performance that is often inferior to the performance of the NLSM, see [LS96b] for details). In the following we also study the rate at which the exact CRB approaches the As CRB, as \( N \) increases. In particular we show that the difference between the exact and the asymptotic CRBs vanishes in a rather fast manner with increasing \( N \).
8.4.1 Estimation Performance versus SNR

We begin by studying the performance of the NLS estimates, in comparison with the asymptotic CRB, as the SNR varies. In the following the local SNR for the kth cisor band is defined as

$$\text{SNR}_k = 10 \log_{10} \frac{\alpha_k^2}{\phi(\omega_k)} \text{ [dB]}.$$  \quad (8.4.37)

In the figures we show the first local SNR value for the scenario under consideration. The data studied consists of two complex sinusoids with unit amplitude, located at the frequencies $f_1 = 0.2106$ and $f_2 = 0.2262$, corrupted by colored noise. The noise is generated by passing a complex Gaussian white noise, with zero mean and unit variance, through a second order auto-regressive (AR) coloring filter given by $1/(1 - 0.1z^{-1} + 0.3z^{-2})$. The two cisors have the initial phases $\phi_1 = \pi/10$ and $\phi_2 = \pi/5$. Before being used in the RELAX algorithm, the $N = 128$ data points are zeropadded to 16 times the data length to obtain a reasonably fine grid on the frequency axis (see [LS96b] for details). The standard deviations of the estimation errors, obtained from 100 Monte Carlo simulations, are plotted in Figures 8.1(a)–(c) together with (the square root of) the respective asymptotic CRB. We show results for the first cisor only (the estimates for the second cisor behaved similarly, and are thus omitted in the interest of brevity).

As can be seen from Figures 8.1(a)–(c), the variance of the estimates is close to the asymptotic CRB for a wide range of SNR values. The threshold SNR (below which the estimated accuracy degrades rapidly) is about -8 dB in the present case. It is also worth noting that the accuracy cannot be increased significantly beyond a certain SNR, in this case about 5 dB. This is related to the resolution on the frequency axis afforded by the zeropadding considered (indeed the accuracy can be improved by increasing the zeropadding length and hence diminishing the quantization effects induced by the sampling of the frequency variable). This clearly leads to a trade-off between the accuracy of the estimates and the amount of calculations needed. The Figures 8.2(a)–(c) show the standard deviations of the estimation errors, together with the respective asymptotic CRB, when the data is zeropadded to 32 times the data length. Generally speaking, the zeropadding length, $M$ say, should be such that $1/M$ is much smaller than the standard deviation of the frequency estimates. We have also empirically found that the smaller the frequency separation between two cisors, the larger the $M$ needed.
Figure 8.1: The standard deviation of the (a) frequency, (b) amplitude and (c) phase estimate as SNR varies (solid line), compared with the corresponding asymptotic CRB (dashed line). Data is zeropadded to 16 times the data length ($N = 128$).
Figure 8.2: The standard deviation of the (a) frequency, (b) amplitude and (c) phase estimate as SNR varies (solid line), compared with the corresponding asymptotic CRB (dashed line). Data is zeropadded to 32 times the data length ($N = 128$).
8.4.2 Estimation Performance versus Sample Length

Next we study the effect of the data length, $N$, on the performance of the NLS estimates and the CRB. If not stated otherwise, the data parameters are as described in the above example. First, we let $N$ vary from $N = 10$ to $N = 200$ with the purpose of studying the rate at which the exact CRB in [FF95] (see also equations (8.2.10)–(8.2.14) of the present paper) approaches the asymptotic CRB in (8.2.28). Figures 8.3(a)–(c) display the exact and asymptotic CRB standard deviations for the first cisoid, as well as the empirical standard deviations of the NLS (as obtained from 100 Monte Carlo simulations). The figures show that the exact CRB converges to the asymptotic CRB in a rather fast manner: the two bounds practically coincide for $N \geq 80$. They also show that in the present case the “threshold value” of $N$ is about $N = 40$: under this value the empirical standard deviation of NLS tends to increase abruptly and takes on values that make the estimate more or less useless. For $N < 40$ neither bound provides a good approximation to the empirical NLS standard deviation; for $40 \leq N \leq 80$ the NLS standard deviation is better approximated by the asymptotic CRB for the amplitude estimate and by the exact CRB for the frequency and phase estimates. Hence the exact CRB is not necessarily a better approximation of the NLS standard deviation in the threshold region, as one might conjecture. Furthermore, note that, for $40 \leq N \leq 80$, the exact CRB does not always bound the standard deviation of the NLS from below presumably owing to the fact that the NLS estimate is biased in this region. For $N \geq 80$ both bounds provide a reasonably good guess for the NLS standard deviation.

Next we consider a case in which we let the second frequency vary as $f_2 = 0.2106 + 2/N$ to obtain a reasonably small frequency separation for all values of $N$ considered. The data are zeropadded to 16 times the data length. The variance of the colored noise is chosen to give a SNR of -5 dB for the first cisoid. Figure 8.4(a)–(c) show the standard deviations of the estimation errors for the first cisoid, as obtained from 100 Monte Carlo simulations, together with the corresponding asymptotic CRB. As seen from these figures, the asymptotic CRB is reached for as few as one hundred data points.
Figure 8.3: The empirical standard deviation of the NLS (a) frequency, (b) amplitude and (c) frequency estimate versus $N$ (dotted line), compared with the exact CRB (solid line) and the asymptotic CRB (dashed line). Data is zeropadded to 64 times the data length; SNR=0 dB.
Figure 8.4: The standard deviation of the (a) frequency, (b) amplitude and (c) phase estimate as the data length varies (solid line), compared with the corresponding asymptotic CRB (dashed line). Data is zeropadded to 16 times the data length (SNR = -5 dB).
8.4.3 Estimation of the Asymptotic CRB

In practice it is always of interest to assess the accuracy of a parameter estimation exercise. In what follows we show that for the NLS estimate we can assess the estimation accuracy based on the available data solely by estimating the asymptotic CRB. We use the estimated amplitudes, phases and frequencies to estimate the additive noise sequence. The estimated noise is then used to estimate the coloring AR filter parameters. To this end we used the forward-backward least-squares approach [Kay88], but other approaches should give similar results. The spectral density of the noise is readily calculated from the estimated AR filter parameters, thus enabling an estimation of the asymptotic CRB by using (8.2.28).

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image1.png}
\caption{(a)}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image2.png}
\caption{(b)}
\end{figure}

**Figure 8.5:** The estimated asymptotic CRB for the frequency estimate as the SNR varies (solid line), compared with the corresponding true asymptotic CRB (dashed line) and the NLS’s empirical standard deviation (stars). Data is zeropadded to (a) 16 (b) 64 times the data length ($N = 64$).
In the next example we consider two cisoids with unit amplitude, frequencies \( f_1 = 0.2106 \) and \( f_2 = 0.2419 \) and initial phases \( \phi_1 = \pi/10 \) and \( \phi_2 = \pi/5 \). The corrupting noise sequence is as in the first example above. The data length is \( N = 64 \) and the SNR is varying. The data is initially zeropadded to 16 times the data length. Figure 8.5(a) shows the estimated asymptotic CRB for the first frequency estimate, together with the respective true asymptotic CRB and the empirical standard deviation of the NLS estimate, as computed from 100 Monte Carlo simulations.

As in the previous examples, one can see how the length of the zeropadding limits the accuracy of the estimate in the high SNR region. By extending the zeropadding length, a better accuracy can be attained. Figure 8.5(b) shows the results when the data is zeropadded to 64 times the data length, which clearly increases the accuracy for higher SNRs.

### 8.4.4 Estimation Performance for Real Radar Data

Finally, we apply the NLS method to experimental data measured by a ground-to-air radar when the target was an aircraft. According to prior studies (see [LS96b] for further details), the measured data, which consists of 64 data points, can be well modeled as a sum of ten cisoids in additive third-order AR noise.

![Figure 8.6: The NLS estimate of the amplitude-to-frequency plot compared with the periodogram, for the radar data. Data was zeropadded to 16 times the data length (\( N = 64 \)).](image)

We used the RELAX algorithm of [LS96b] to obtain the NLS estimates of the parameters of the ten sinusoidal components in the data under study. The amplitude-versus-frequency plot so estimated is shown in Figure 8.6, where we also show the periodogram of the data at hand for comparative purposes. We remark on the fact that the periodogram fails to resolve the closely-spaced spectral lines at about \((f_2 = 0.11, f_3 = 0.12)\) and \((f_7 = 0.21, f_8 = 0.23)\), which are revealed by the NLS. Figures 8.7(a) and (b)
Table 8.1: The (square root of the) estimated asymptotic CRB for the radar data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.5 \times 10^{-1}</td>
<td>1.2 \times 10^{-3}</td>
<td>9.3 \times 10^{-3}</td>
</tr>
<tr>
<td>2</td>
<td>1.4 \times 10^{-5}</td>
<td>2.6 \times 10^{-1}</td>
<td>5.2 \times 10^{-4}</td>
</tr>
<tr>
<td>3</td>
<td>3.5 \times 10^{-5}</td>
<td>3.5 \times 10^{-1}</td>
<td>1.3 \times 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>1.4 \times 10^{-1}</td>
<td>8.7 \times 10^{-1}</td>
<td>5.2 \times 10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>4.2 \times 10^{-5}</td>
<td>5.1 \times 10^{-1}</td>
<td>1.6 \times 10^{-3}</td>
</tr>
<tr>
<td>6</td>
<td>4.3 \times 10^{-5}</td>
<td>4.4 \times 10^{-1}</td>
<td>1.6 \times 10^{-3}</td>
</tr>
<tr>
<td>7</td>
<td>1.7 \times 10^{-5}</td>
<td>3.1 \times 10^{-1}</td>
<td>6.4 \times 10^{-4}</td>
</tr>
<tr>
<td>8</td>
<td>1.2 \times 10^{-1}</td>
<td>9.4 \times 10^{-1}</td>
<td>4.4 \times 10^{-3}</td>
</tr>
<tr>
<td>9</td>
<td>1.1 \times 10^{-4}</td>
<td>8.1 \times 10^{-1}</td>
<td>4.0 \times 10^{-3}</td>
</tr>
<tr>
<td>10</td>
<td>1.6 \times 10^{-4}</td>
<td>1.0 \times 10^{-3}</td>
<td>6.0 \times 10^{-3}</td>
</tr>
</tbody>
</table>

show the quality of the estimation, by plotting the real and imaginary parts of the true data compared with the real and imaginary parts of the estimated sinusoidal components. The residual error of the estimation is then modeled as a third order AR process, giving the spectral density of the additive noise. The estimated spectral density together with the estimated sinusoidal parameters enable the asymptotic CRB to be estimated by using (8.2.28). The estimated values of As CRB, shown in Table 8.1, indicate an excellent estimation accuracy.
Figure 8.7: The (a) real (b) imaginary part of the true data (solid line) compared with the corresponding part of the estimated sinusoidal component (dashed line).
Chapter 9

On the Identifiability of Multipath Parameters

Consider a received signal consisting of noise and an unknown waveform that arrives at a sensor through both a direct path and an attenuated and delayed specular path. We show that a commonly-used frequency-domain model for such a signal is valid essentially only under the assumption that the waveform that underwent multipath propagation is band-limited. Furthermore, we derive sufficient conditions for the aforementioned model to be identifiable from second-order statistics (i.e., its parameters can be uniquely determined from the spectral density of the received signal). We also show that in the frequently considered case of a white waveform, neither the attenuation coefficient, nor the waveform power, nor the noise power can be unambiguously determined.

9.1 Multipath Parameter Estimation Problem

Let $x(t)$, for $t \in \mathbb{R}$, denote a complex-valued continuous-time waveform that arrives at a sensor via both a direct path and a specular path. The observed discrete-time signal $y(k)$, for $k \in \mathbb{Z}$, can then be written as:

$$
y(k) = x(k\Delta) + gx((k - \tau)\Delta) + e(k),
$$

for $k = 0, 1, 2, \ldots$, where $\Delta$ is the sampling period, $g$ is the attenuation coefficient ($g \in \mathbb{C}$), $\tau \Delta$ is the time delay ($\tau \in \mathbb{R}$), and $e(k)$ is (discrete-time) observation noise. We assume that $x(t)$ is stationary and that $e(k)$ is complex-valued white noise, uncorrelated with $x(k\Delta)$, with mean zero and variance denoted by $\sigma_e^2$. 
The multipath estimation problem consists of estimating $\tau, g, \sigma^2_e$ and the parameters of any model postulated for $x(k\Delta)$, from a set of observations $\{y(k)\}$. There is a rich literature on this problem and a host of methods for solving it (see, e.g., [Car93] and the many references therein). These methods find application particularly in radar and sonar. Recently there has been a renewed interest in the multipath parameter estimation problem owing to its applications in wireless communications [Swi98, JSS98b, ASS81].

Let $\Phi_y(\omega)$, for $\omega \in [-\pi, \pi]$, denote the power spectral density (PSD) of $y(k)$. A frequency-domain model often used to derive estimates of the parameters in (9.1.1) is the following (see, e.g., [SF85, CRP80]):

$$
\Phi_y(\omega) = |1 + ge^{i\omega\tau}|^2 \Phi_x(\omega) + \sigma^2_e, \tag{9.1.2}
$$

where $\Phi_x(\omega)$ is the PSD of the sequence $\{x(k\Delta)\}$. However, the above equation for $\Phi_y(\omega)$ is not always valid. Indeed, $\Phi_y(\omega)$ in (9.1.2) is completely determined by the multipath parameters and the PSD of $x(k\Delta)$; however, because $\tau$ is not in general an integer, the PSD of $y(k)$ should also depend on the properties of the inter-sample values of $x(t)$.

A first goal of this short chapter is to show that (9.1.2) holds if and (essentially) only if $x(t)$ is band-limited to $\pi/\Delta$; that is, the PSD of the continuous-time waveform $x(t)$ is zero outside the interval $[-\pi/\Delta, \pi/\Delta]$. This result, the intuitive reason for which is that under the previous condition $x(t)$ can be exactly retrieved from $x(k\Delta)$, is apparently known. Even so, a proof of it is not readily available in the literature. We provide such a proof in Section 9.2.

Next, in Section 9.3 we analyze the identifiability properties of the PSD model in (9.1.2). We derive sufficient conditions under which $\tau, g, \sigma^2_e$ and the parameters of a model postulated for the discrete-time signal $x(k\Delta)$ can be uniquely determined from $\Phi_y(\omega)$. We also show that in the frequently considered case of a white $x(k\Delta)$ sequence, $\tau$ can still be uniquely determined, but $g, \sigma^2_e$ and $\sigma^2_d$ cannot.

### 9.2 Derivation of the PSD Model

Let

$$
r(k) = E[y(j)y^*(j + k)], \quad k \in \mathbb{Z}
$$

where $(\cdot)^*$ denotes the complex conjugate, and let

$$
\rho(s) = E[x(t)x^*(t + s\Delta)], \quad s \in \mathbb{R}.
$$

A simple calculation shows that $r(k)$ can be rewritten as

$$
r(k) = E[x(j\Delta) + gx((j - \tau)\Delta) + e(j)] \times
$$
\[
\frac{x((j + k)\Delta) + gx((j + k - \tau)\Delta) + e(j + k)]^*}{(1 + |g|^2) \rho(k) + e^* \rho(k - \tau) + \rho(k + \tau) + \sigma_e^2 \delta_{k,0}}
\]  
(9.2.3)

and hence that
\[
\Phi_y(\omega) = \sigma_e^2 + (1 + |g|^2) \sum_{k = -\infty}^{\infty} \rho(k)e^{-i\omega k} + e^* \rho(k - \tau) + \sum_{k = -\infty}^{\infty} \rho(k + \tau)e^{-i\omega k},
\]  
(9.2.4)

for \( \omega \in [-\pi, \pi] \) in (9.2.3), \( \delta_{k,0} = 1 \) for \( k = 0 \) and \( \delta_{k,0} = 0 \) for \( k \neq 0 \). If
\[
\sum_{k = -\infty}^{\infty} \rho(k \pm \tau)e^{-i\omega k} = e^{\pm i\omega \tau} \sum_{k = -\infty}^{\infty} \rho(k)e^{-i\omega k}
\]
\[
\triangleq e^{\pm i\omega \tau} \Phi_x(\omega)
\]  
(9.2.5)

were true, then (9.1.2) would be readily obtained by inserting (9.2.5) in (9.2.4). Hence our task is to study the conditions under which (9.2.5) holds.

Under the assumption that the PSD of \( x(t) \) is band-limited to \( \pi/\Delta \), we have (by Shannon’s interpolation formula):
\[
\rho(s) = \sum_{k = -\infty}^{\infty} \rho(k) \text{sinc} ((s - k)\Delta)
\]  
(9.2.6)

where
\[
\text{sinc}(x) \triangleq \frac{\sin(\pi x/\Delta)}{\pi x/\Delta}
\]  
(9.2.7)

(note that a relationship similar to (9.2.6) relates \( x(t) \) to \( x(k\Delta) \)). By the same token,
\[
e^{i\omega \Delta s} = \sum_{k = -\infty}^{\infty} e^{i\omega \Delta k} \text{sinc} ((s - k)\Delta)
\]  
(9.2.8)

for \( |\omega| \leq \pi \). Making use of (9.2.6) and (9.2.8), we obtain:
\[
\sum_{k = -\infty}^{\infty} \rho(k + \tau)e^{-i\omega k} = \sum_{k = -\infty}^{\infty} \sum_{p = -\infty}^{\infty} \rho(p) \text{sinc} ((k + \tau - p)\Delta)e^{-i\omega k}
\]
\[
= \sum_{p = -\infty}^{\infty} \rho(p) \sum_{k = -\infty}^{\infty} e^{-i\frac{\pi}{\Delta} k} \text{sinc} ((p - \tau - k)\Delta)
\]
\[
= \sum_{p = -\infty}^{\infty} \rho(p)e^{-i\omega(p - \tau)}
\]
\[
e^{i\omega \tau} \Phi_x(\omega)
\]  
(9.2.9)
which proves (9.2.5), and hence concludes the derivation of (9.1.2) under the assumptions stated.

If the PSD of \( x(t) \) is not band-limited to \( \pi/\Delta \), then in general (9.2.6) will not hold exactly and this will induce an error in (9.2.9) and hence in (9.1.2). Quantification of this error, for instance as a function of the power of \( x(t) \) outside the band \([ -\pi/\Delta, \pi/\Delta ] \) is an interesting problem that is however beyond the scope of this short study. As a matter of fact, the band-limitation of \( x(t) \) to \( \pi/\Delta \) is a fairly weak condition that can be enforced, for instance, by prefiltering the observed data through a low-pass filter with cut-off frequency less than \( \pi/\Delta \).

### 9.3 Identifiability Properties of the PSD Model

In what follows we assume that the sequence \( \{ x(k\Delta) \} \) is an auto-regression (AR) of order \( n \), which implies that

\[
\Phi_x(\omega) = \frac{\sigma_w^2}{|A(e^{j\omega})|^2} \tag{9.3.10}
\]

for some \( \sigma_w^2 > 0 \) and some monic (real) polynomial \( A(z) \) of degree \( n \) that has all zeros strictly inside the unit circle. Since any continuous PSD \( \Phi_x(\omega) \) can be approximated as well as desired by the right-hand-side of (9.3.10) with a sufficiently large value of \( n \), the above assumption on \( x(k\Delta) \) is not too restrictive. We also assume that \( n \) is given.

The analysis of the identifiability properties of (9.1.2) under (9.3.10) reduces to studying the solution set \( \{ \hat{\tau}, \hat{\theta}, \sigma_c, \sigma_w, \hat{A}(e^{j\omega}) \} \) of the equation :

\[
\hat{\sigma}_c^2 + \left| 1 + \hat{\theta} e^{j\omega} \right|^2 \frac{\hat{\sigma}_w^2}{\left| \hat{A}(e^{j\omega}) \right|^2} = \sigma_c^2 + \left| 1 + \theta e^{j\omega} \right|^2 \frac{\sigma_w^2}{\left| A(e^{j\omega}) \right|^2} \tag{9.3.11}
\]

for \( |\omega| \leq \pi \). The PSD model (9.1.2) is identifiable if and only if the unique solution to (9.3.11) is \( \hat{\tau} = \tau \), \( \hat{\theta} = \theta \), etc.

Let

\[
\left| A(e^{j\omega}) \right|^2 \triangleq \alpha_0 + \alpha_1 \cos(\omega) + \ldots + \alpha_n \cos(n\omega). \tag{9.3.12}
\]

Equation (9.3.11) implies that

\[
\left| \hat{A}(e^{j\omega}) \right|^2 = \left| A(e^{j\omega}) \right|^2 \tag{9.3.13}
\]

whose unique solution is (by the spectral factorization theorem) :

\[
\hat{A}(e^{j\omega}) = A(e^{j\omega}) \tag{9.3.14}
\]
(note that to obtain (9.3.13) from (9.3.11) we made use of the fact that there is no possible cancellation of factors in \( |1 + g e^{i \omega \tau}|^2 / |A(e^{i \omega})|^2 \) since the denominator is bounded away from zero for any \( \omega \), by assumption). Inserting (9.3.13) into (9.3.11) yields the equation:

\[
(\hat{\sigma}_c^2 - \sigma_c^2) [\alpha_0 + \alpha_1 \cos(\omega) + \ldots + \alpha_n \cos(n\omega)] +
\left[ 1 + |\hat{g}|^2 + 2g_r \cos(\omega \hat{\tau}) - 2g_i \sin(\omega \hat{\tau}) \right] \hat{\sigma}_w^2 -
\left[ 1 + |g|^2 + 2g_r \cos(\omega \tau) - 2g_i \sin(\omega \tau) \right] \sigma_w^2 = 0,
\]

(9.3.15)

where \( g_r \) and \( g_i \) denote the real and imaginary parts of \( g \). The functions \( \cos(\omega), \ldots, \cos(n\omega) \) in (9.3.15) are linearly independent. We assume that at least two linear combinations of the functions \( \cos(\omega), \ldots, \cos(n\omega) \), \( \cos(\omega \tau) \) are linearly independent, which is a weak assumption that holds if either \( \tau = \text{non-integer} \) and \( n \geq 1 \), or \( \tau = \text{integer larger than } n \geq 1 \), or \( \tau = \text{integer less than } n \geq 3 \). Under this assumption (9.3.15) is equivalent to:

\[
\hat{\sigma}_c^2 = \sigma_c^2 \quad (9.3.16)
\]

\[
\hat{\tau} = \tau \quad (9.3.17)
\]

and

\[
(1 + |\hat{g}|^2) \hat{\sigma}_w^2 = (1 + |g|^2) \sigma_w^2 \quad (9.3.18)
\]

\[
\angle \hat{g} = \angle g \quad (9.3.19)
\]

\[
|g| \sigma_w^2 = |\hat{g}| \sigma_w^2 \quad (9.3.20)
\]

(above we have also assumed that \( g \sigma_w^2 \neq 0 \), otherwise the delay \( \tau \) is obviously not identifiable).

Next we divide the equations in (9.3.18) and (9.3.20) member by member to obtain:

\[
0 = (1 + |\hat{g}|^2) |g| - (1 + |g|^2) |\hat{g}|
= (|g| - |\hat{g}|)(1 - |g||\hat{g}|). \quad (9.3.21)
\]

This equation shows that the true solution \( \hat{g} = g \) and the false one \( |g| = 1/|g| \) are indistinguishable from second-order data statistics. Usually one can assume that

\[
|g| < 1 \quad (9.3.22)
\]

which helps choosing the desired solution

\[
\hat{g} = g. \quad (9.3.23)
\]

Insertion of (9.3.23) into (9.3.20) yields

\[
\hat{\sigma}_w^2 = \sigma_w^2. \quad (9.3.24)
\]
To summarize, the PSD model (9.1.2) is identifiable under the weak assumption that the sequence \( \{x(k\Delta)\} \) is an auto-regression of order \( n \geq 1 \), \( \tau = \text{non-integer} \) (most likely) or integer larger than \( n \) (if \( \tau \) is an integer less than \( n \) then we need \( n \geq 3 \)), and the attenuation coefficient \( g \) satisfies (9.3.22).

The previous identifiability condition set excludes the case of \( n = 0 \), which is somewhat intriguing since in the literature the sequence \( \{x(k\Delta)\} \) has often been assumed to be white (see, e.g., [CRP80]). In such a case, equation (9.3.15) reduces to

\[
\hat{\tau} = \tau \quad (9.3.25)
\]

\[
|g|\sigma_w^2 = |g|\sigma_w^2 \quad (9.3.26)
\]

\[
\sigma_c^2 + (1 + |g|^2)\sigma_w^2 = \sigma_c^2 + (1 + |g|^2)\sigma_w^2. \quad (9.3.27)
\]

Hence \( \tau \) is identifiable. However the other parameters \( g, \sigma_c^2 \) and \( \sigma_w^2 \) are not. As an example, let

\[
g = \sqrt{0.1}, \quad \sigma_w^2 = 1.
\]

Then, it can be easily checked that

\[
\hat{g} = 2\sqrt{0.1}, \quad \hat{\sigma}_w^2 = 1/2
\]

and

\[
\hat{\sigma}_c^2 = \sigma_c^2 + 0.4
\]

is a possible false solution to (9.3.26) and (9.3.27). Many other false solutions exists (note that we only have two equations, (9.3.26) and (9.3.27), to determine three unknown parameters, \( \hat{g}, \hat{\sigma}_c^2 \) and \( \hat{\sigma}_w^2 \)).

Finally, note that it is presently unknown whether the above identifiability result, stated following equation (9.3.24), is extendable to auto-regressive moving average (ARMA) sequences. Until the question about such an extension is resolved we should probably trade-off computational complexity for identifiability and use an AR model for \( x(k\Delta) \) even when it is suspected that this sequence can be described by an ARMA equation with a number of parameters that is much smaller than \( n \).
Chapter 10

Subspace-Based Estimation of Time Delays and Doppler Shifts

This chapter considers the problem of estimating the time delays and Doppler shifts of a known waveform received via several distinct paths by an array of antennas. The general maximum likelihood estimator is presented, and is shown to require a $2d$-dimensional non-linear minimization, where $d$ is the number of received signal reflections. Two alternative solutions based on signal and noise subspace fitting are proposed, requiring only a $d$-dimensional minimization. In particular, we show how to decouple the required search into a two-step procedure, where the delays are estimated and the Dopplers solved for explicitly. Initial conditions for the time delay search can be obtained by applying generalizations of the MUSIC and the ESPRIT algorithms, which are also outlined in the chapter. Simulation examples are included to illustrate the algorithms' performance relative to the Cramér-Rao bound.

10.1 Introduction

The problem of using an antenna array to estimate the time delays and Doppler shifts (or frequency offsets) of a known signal is important in two common applications. First, in active radar and sonar, a known waveform is transmitted and reflections from objects “illuminated” by the transmission are subsequently received. The received signals are often modeled as scaled, delayed, and Doppler shifted versions of the transmitted signal. Estimation
of the signal amplitude, delay, and Doppler shift provides information about the position and relative motion of the objects [Lev88, Edd93].

The second application involves estimation of the parameters of a multipath communication channel in situations where the transmitter is rapidly moving or has an unknown frequency offset. For example, consider a situation where a remote mobile user transmits a known waveform (e.g., a training sequence) to a base station for synchronization or equalization purposes. If the channel is frequency selective (non-zero delay spread), then the signal will be received with several distinct delays. In addition, due to the motion of the mobile and variations in the carrier frequency of the transmitter, the known signal can also be received with a small frequency offset. Estimation of the delays and frequency offsets, as well as the spatial signatures of the signal arrivals, is necessary in establishing a clean, inter-symbol and interference-free communication link.

This chapter presents a novel approach to solving the problems described above. The techniques presented are applicable in situations involving multiple antennas and, unlike classical methods, are asymptotically optimal at high signal-to-noise ratio (SNR) even when multiple overlapping copies of the signal are received. The frequency domain model used in [Swi95, Swi98] for time delay estimation is generalized to incorporate the presence of (small) frequency offsets. The resulting signal manifold in the frequency domain is shown to be a generalized version of the signal manifold of [Swi95, Swi98], in much the same way that polarization [Sch81, FP83, SV93] and local scattering [AOS97, Ast99] generalize the standard array manifold in direction of arrival (DOA) estimation. This observation motivates the development of subspace-based techniques similar to those in [FP83, SV93, AOS97], which provide closed-form solutions for the linear parameters (in our case, the frequency/Doppler offsets). The resulting algorithms require a search for the time delays, but it is shown that for small frequency offsets, the closed-form time delay estimation techniques of [Swi95, Swi98] provide excellent initial conditions.

Classical approaches to time delay and Doppler estimation are based on matched filtering (see, e.g., [Tre68, He95]). These techniques typically assume one signal path and one sensor, although the extension to multiple sensors is straightforward. Matched filtering techniques are known to be optimal in the maximum likelihood (ML) sense for a single signal arrival, but are not consistent when multiple overlapping copies of the signal are present. While a number of authors have proposed time delay estimators that exploit frequency domain data models, their use in Doppler estimation has not been widespread. When such models have been used, they have again only focused on the single signal path case [Wax82, Ste93]. Other
recently proposed techniques for the case of a single signal arrival include the wideband ambiguity function method of [JWL95], and the structured covariance estimator of [CB96]. A recent paper [HVK97] presents a deconvolution approach for resolving multiple delayed and Doppler shifted paths, but only over a quantized parameter grid. The key features of the methods proposed below are that

(1) they provide continuous-valued estimates of time delays and Doppler shifts for multiple signal arrivals, and

(2) they are parametric estimators with asymptotic accuracy equivalent to that of the maximum likelihood approach.

The outline of the chapter is as follows. In the next section, we present time and frequency domain versions of the data model assumed in this work. By interchanging the roles of the samples in space and time, we show how the time delay and Doppler estimation problem can be cast in the well-studied framework of DOA estimation. In particular, we draw parallels between the array manifold in space that arises in DOA estimation, and the signal manifold in time that we employ in this work. Under this paradigm, the classical matched filtering approach is seen to be equivalent to the simple delay-and-sum beamformer. Section 10.3 then presents the ML solution to the multiple sensor, multiple signal arrival problem, and outlines the corresponding Cramér-Rao bound. The ML solution is shown to require, in general, a search over both the delay and Doppler parameters, and may thus be difficult to implement in practice. For this reason, two asymptotically equivalent (in SNR) subspace fitting algorithms are derived in Section 10.4. These algorithms exploit the fact that, to first order, the signal in the frequency domain depends on the Doppler frequency in a linear fashion, and hence the Doppler may be estimated explicitly. A search is still required for the time delays, but this can be conveniently initialized by two suboptimal algorithms based on MUSIC [Sch81] and ESPRIT [RK89] that are also presented. Finally, Section 10.5 provides the results of a number of simulation examples to illustrate the relative performance of the proposed algorithms.
10.2 Modeling

10.2.1 Assumptions

Suppose an $m$-element antenna array receives several scaled, time-delayed, and frequency/doppler-shifted copies of a known baseband signal, $s(t)$. The received signals could, for instance, be the echoes from a pulse transmitted by an active radar, or they could result from a training sequence sent over a multipath communication channel. In either case, we may model the output of the array for small frequency/Doppler offsets as

$$x(t) = \sum_{k=1}^{d} a_k s(t - \tau_k) e^{j\omega_D t} + n(t), \quad (10.2.1)$$

where $d$ represents the number of different multipath signals, and where the parameters $\tau_k$, $\omega_D$, and $a_k$ are the time delay, frequency offset, and spatial signature of the $k$th arrival. The additive noise vector, $n(t)$, is assumed to be a zero mean temporally and spatially white noise process with covariance $\sigma^2 I$. The standard narrowband assumption is employed here, i.e., the propagation time of the signal across the array is assumed to be much less than the reciprocal of the signal bandwidth. Note that, for the radar case, the frequency offset $\omega_D$ is a narrowband approximation to the stretching or shrinking of the frequency axis due to the Doppler effect induced by the relative motion of the reflecting target (see also the discussion in [JS98]).

The model in (10.2.1) could be further parameterized in terms of a set of DOAs. For example, if we let $a(\theta)$ represent the array response to a unit amplitude plane wave arriving from the DOA $\theta$, then $a_k$ might be written as

$$a_k = \sum_{i=1}^{d_k} \alpha_{ik} a(\theta_{ik}), \quad (10.2.2)$$

where $\alpha_{ik}$ and $\theta_{ik}$ denote the complex amplitudes and DOAs associated with the $i$th arrival in the $k$th cluster of echoes that have the same delay and Doppler shift. Here, to simplify the problem, we do not use this explicit parameterization of the spatial response in terms of DOAs, but instead treat the elements of $a_k$ as deterministic parameters to be estimated. This allows us to consider a cluster of coherent arrivals that share a given time delay and Doppler shift, without the necessity of estimating the number of such arrivals nor their individual DOAs and amplitudes. After an estimate $\hat{a}_k$ is obtained, $d_k$, $\alpha_{ik}$, and $\theta_{ik}$ could be estimated if desired by using a least-squares fit of the model in (10.2.2).
Assuming that $\mathbf{x}(t)$ is an $m \times 1$ column vector, and that a total of $N$ snapshots are collected from the array, the data may be arranged in matrix form as

$$
\mathbf{X}_t \triangleq \begin{bmatrix}
\mathbf{x}^T(t_1) \\
\vdots \\
\mathbf{x}^T(t_N)
\end{bmatrix} = \left( \mathbf{S}_t(\tau) \odot \mathbf{V}_t(\omega) \right) \mathbf{A} + \mathbf{N}_t \tag{10.2.3}
$$

$$
\triangleq \mathbf{Q}_t(\tau, \omega) \mathbf{A} + \mathbf{N}_t , \tag{10.2.4}
$$

where $\mathbf{N}_t$ is formed in the same way as $\mathbf{X}_t$, $\odot$ denotes the Schur-Hadamard product, and

$$
\tau = [\tau_1 \ldots \tau_d]^T \\
\omega = [\omega_{D_1} \ldots \omega_{D_d}]^T \\
\mathbf{A} = [a_1 \ldots a_d]^T \\
\mathbf{S}_t(\tau) = [s_t(\tau_1) \ldots s_t(\tau_d)] \\
\mathbf{s}_t(\tau) = [s(t_1 - \tau) \ldots s(t_N - \tau)]^T \\
\mathbf{V}_t(\omega) = [\mathbf{v}_1(\omega_{D_1}) \ldots \mathbf{v}_1(\omega_{D_d})] \\
\mathbf{v}_t(\omega) = [e^{j\omega_{T_1}} \ldots e^{j\omega_{T_N}}]^T .
$$

The columns of the matrix $\mathbf{Q}_t(\tau, \omega)$ are given by

$$
\mathbf{q}_t(\tau_k, \omega_{D_k}) = \mathbf{s}_t(\tau_k) \odot \mathbf{v}_t(\omega_{D_k}) .
$$

The subscript $t$ is used to distinguish the above time-domain model from its frequency domain counterpart presented below.

The frequency domain representation of the array output in (10.2.1) is given by

$$
\mathbf{x}(\omega) = \sum_{k=1}^{d} \mathbf{a}_k s(\omega - \omega_{D_k}) e^{-j\omega \tau_k} + \mathbf{n}(\omega) , \tag{10.2.5}
$$

where $\mathbf{x}(\omega)$, $s(\omega)$ and $\mathbf{n}(\omega)$ are the Fourier transforms of $\mathbf{x}(t)$, $s(t)$ and $\mathbf{n}(t)$, respectively. Under the assumption that the frequency/Doppler offsets are “small”, it is possible to simplify the dependence of (10.2.5) on the Doppler frequencies by neglecting the higher order terms in the Taylor series expansion of $s(\omega - \omega_{D_k})$:

$$
\begin{bmatrix}
s(\omega_1 - \omega_{D_k}) \\
\vdots \\
s(\omega_N - \omega_{D_k})
\end{bmatrix} \approx \begin{bmatrix}
s(\omega_1) \\
\vdots \\
s(\omega_N)
\end{bmatrix} - \omega_{D_k} \begin{bmatrix}
d(\omega_1) \\
\vdots \\
d(\omega_N)
\end{bmatrix}.$$
\[ \Delta \equiv s - \omega_{D_k} d \]  
\[ d(\omega_i) = \frac{\partial s(\omega_i)}{\partial \omega} \bigg|_{\omega = \omega_i} . \]

As with \( x(t) \), we assume that \( x(\omega) \) is an \( m \times 1 \) column vector, and we collect the array data at frequencies \( \omega_1, \ldots, \omega_N \) in matrix form as

\[ X \triangleq \begin{bmatrix} x^T(\omega_1) \\ \vdots \\ x^T(\omega_N) \end{bmatrix} = \left( \begin{bmatrix} S V(\tau) - D V(\tau) \Phi(\omega) \end{bmatrix} \right) A + N \]  
\[ \triangleq Q(\tau, \omega) A + N, \]  
\[ \Delta \]

where

\[ S = \text{diag}(s) \]
\[ D = \text{diag}(d) \]
\[ V(\tau) = [v(\tau_1) \ldots v(\tau_d)] \]
\[ v(\tau) = [\exp(-j\omega_1\tau) \ldots \exp(-j\omega_N\tau)]^T \]
\[ \Phi(\omega) = \text{diag}(\omega), \]

and where, for example, \( \text{diag}(\omega) \) is a diagonal matrix with the elements of the vector \( \omega \) along its diagonal. The columns of \( Q(\tau, \omega) \) have the following form:

\[ q(\tau_k, \omega_{D_k}) = S v(\tau_k) - \omega_{D_k} D v(\tau_k). \]

In practice, \( X \) is obtained by performing a DFT on the time domain data in \( X_t \). As such, the translation of time delays into a linearly increasing phase shift \( e^{-j\omega\tau} \) does not hold exactly, except in certain special cases involving, for example, a periodic signal or a signal with finite time support. However, if \( t_N - t_1 \gg \max_k \tau_k \) and the signal is sampled at least at the Nyquist rate, then the error induced by the finite length DFT will be small, and the frequency domain model will be a reasonable approximation (this is illustrated by the simulation results in Section 10.5). As explained below, the model in (10.2.8) has some interesting links with the well-known DOA estimation problem.
10.2.2 Connections with DOA Estimation

By interchanging the roles of the samples in time and space, the delay and Doppler estimation problem can be cast into the more familiar framework of DOA estimation. To see this, compare (10.2.8) with the standard model used in DOA estimation (see, e.g., [Boh91, SM97]):

\[ X = A(\theta)S + N, \quad (10.2.9) \]

where \( \theta \) is a vector containing the DOAs of the signals. In (10.2.9), \( A \) is a known function of the \( d \) parameters in \( \theta \), and \( S \) is usually treated as an unknown unstructured matrix. On the other hand, in (10.2.8) it is \( Q \) that is parameterized and \( A \) that is unstructured. In essence, the roles of time (frequency) and space have thus been reversed. Instead of the array manifold \( a(\theta) \) in \( m \)-space employed in the DOA model, the delay/Doppler model uses a “signal” manifold \( q(\tau_k, \omega_{D_k}) \) in \( N \)-space.

A closer parallel may be drawn by comparing (10.2.8) with the generalized array manifold that is associated with polarized antenna arrays [FP83, SV93] and signals with angular spread [AOS97]. For dual polarized arrays, the array manifold is a combination of the response due to each polarization:

\[ A(\theta, \phi_h, \phi_v) = A_h(\theta)\Phi_h + A_v(\theta)\Phi_v, \quad (10.2.10) \]

where \( A_h, A_v \), and the diagonal matrices \( \Phi_h = \text{diag}(\phi_h) \) and \( \Phi_v = \text{diag}(\phi_v) \) represent the array response and the relative contribution of the horizontal and vertical polarization components, respectively. In the angular spread model of [AOS97], the array manifold is given by

\[ A(\theta, \phi) = A(\theta) + D(\theta)\Phi, \quad (10.2.11) \]

where the columns of \( D(\theta) \) are the derivatives of the columns of \( A(\theta) \) with respect to each element of \( \theta \), and \( \Phi = \text{diag}(\phi) \) is a \( d \times d \) diagonal matrix whose elements \( \phi \) are a function of the DOAs and amplitudes of the local scatterers for each source. In [FP83, SV93, AOS97], algorithms were developed which estimate the linear parameters (i.e., the elements of the diagonal matrices denoted by \( \Phi \) above) in closed-form, and require only a search over the \( d \) elements of \( \theta \). In this chapter, a similar approach is proposed for separating out the estimation of \( \omega \) from that of \( \tau \).

A classical approach to DOA estimation is the standard delay-and-sum beamformer, which maximizes

\[ P(\theta) = \sum_{p=1}^{N} |a^\ast(\theta)x(t_p)|^2 \quad (10.2.12) \]
with respect to $\theta$. Equation (10.2.12) can be thought of as a spatial matched filter. Similarly, the classical approach for time delay and Doppler estimation also involves a matched filter that correlates the received signal $x(t)$ with a delayed and frequency shifted version of the known signal [Tre68, Hel95]:

$$
\tau, \omega_D = \arg \max_{\tau, \omega_D} \left| \sum_{p=1}^{N} s^*(t_p - \tau)e^{-j\omega_D t_p} x(t_p) \right|^2
$$

(10.2.13)

$$
= \arg \max_{\tau, \omega_D} \sum_{k=1}^{m} |q_k^*(\tau, \omega_D) X_{t,k}|^2,
$$

(10.2.14)

where $X_{t,k}$ denotes the $k$th column of $X_t$. Both the classical beamforming method in (10.2.12) and the matched filtering approach of (10.2.13) are known to be optimal in the maximum likelihood sense if only a single signal is received ($d = 1$). However, neither (10.2.12) nor (10.2.13) is consistent when multiple arrivals are present. In Section 10.3, we present several methods that overcome the drawbacks of (10.2.13) in the multiple echo case. These methods are counterparts to the subspace fitting [SV93, SS90, VO91, OVS93], MUSIC [Sch81, FP83], and ESPRIT [RK89] algorithms developed for DOA estimation.

### 10.2.3 Identifiability

The parameters of the model in (10.2.4) are said to be identifiable if

$$
Q_l(\tau, \omega) A \neq Q_l(\tau', \omega') A'
$$

(10.2.15)

whenever $\tau \neq \tau'$, $\omega \neq \omega'$ or $A \neq A'$. In other words, the unknown parameters $\tau, \omega,$ and $A$ should be uniquely determinable from noise free data. We use the time-domain signal manifold here since it does not rely on any approximations, such as the one used in (10.2.6) for the frequency domain model. Because of the similarities between (10.2.4) and (10.2.9), the identifiability results of [WZ89] for DOA estimation can be applied here. Our model employs a signal manifold $q_l(\tau, \omega)$ in $N$-space instead of the array manifold $a(\theta)$ in $m$-space. As with $a(\theta)$, the signal's temporal signature $q_l(\tau, \omega)$ is said to be unambiguous if every collection of $N$ distinct vectors from the signal manifold is linearly independent. Reversing the role of time and space, we obtain the following theorem:

**Theorem 10.1** Suppose an $m$-element array receives $d$ delayed and frequency/doppler-shifted copies of a known waveform. If the signal manifold
is unambiguous, then the delays $\tau$, the frequency shifts $\omega$, and the spatial signatures in $A$ can be uniquely determined provided that

$$d < \frac{N + m'}{2},$$  \hspace{1cm} (10.2.16)

where $m' = \text{rank}(A)$. If instead

$$d < \frac{m'}{m' + 1} N,$$  \hspace{1cm} (10.2.17)

then $\tau$, $\omega$, and $A$ may be uniquely determined with probability one.

**Proof:** The proof is identical to that given in [WZ89] for the case where there are two parameters per source (e.g., as in azimuth and elevation DOA estimation).

It is reasonable to assume that in most situations, the matrix $A$ will be full rank so that $m' = \min(m, d)$. However, $A$ may be ill-conditioned in situations where signals with different time delays or Doppler shifts arrive with nearly coincident DOAs. When $m' = d$, identifiability is guaranteed provided that $d < N$. In the case when $m' = m$, the upper bound in (10.2.16) approaches $d < N/2$ for large $N$. In either case, the number of resolvable signal paths is only limited by the amount of data collected from the array.

### 10.3 ML Estimation and the CRB

Assuming temporally and spatially white circular Gaussian noise, and treating $A$ as an unstructured deterministic matrix, the ML solution is easily shown to be equivalent to the following least-squares problem:

$$\hat{\tau}, \hat{\omega}, \hat{A} = \arg \min_{\tau, \omega, A} V_{ML}(\tau, \omega, A)$$

$$= \arg \min_{\tau, \omega, A} \|X - Q(\tau, \omega)A\|_F^2,$$ \hspace{1cm} (10.3.18)

where $\|\cdot\|_F$ denotes the Frobenius norm. Note that (10.3.18) could be used with the time domain data $X_t$ and manifold $Q_t(\tau, \omega)$ as well. The loss function in (10.3.18) can be explicitly minimized with respect to $A$, which yields

$$\hat{A} = Q^\dagger(\hat{\tau}, \hat{\omega})X,$$ \hspace{1cm} (10.3.19)
where
\[
\hat{\tau}, \hat{\omega} = \arg \min_{\tau, \omega} \| \Pi_Q^{\perp}(\tau, \omega) X \|_F^2
\]
\[
= \arg \max_{\tau, \omega} \text{tr} \left\{ X^* \Pi_Q(\tau, \omega) X \right\},
\]
and where $\text{tr} \{ \cdot \}$ denotes the matrix trace, $(\cdot)^*$ the conjugate transpose, $(\cdot)^\dagger$ the pseudo inverse, and
\[
\Pi_Q^{\perp}(\tau, \omega) = I - \Pi_Q(\tau, \omega) = I - Q(\tau, \omega) Q^\dagger(\tau, \omega)
\]
is the orthogonal projection matrix onto the null space of $Q^*(\tau, \omega)$.

Unlike the (deterministic) ML estimator for the DOA problem, the ML estimates obtained from (10.3.20) and (10.3.19) will be consistent and asymptotically efficient, i.e., the estimates will asymptotically (as $N \to \infty$) achieve the Cramer-Rao lower bound (CRB). This results since $Q^\dagger(\hat{\tau}, \hat{\omega})$ in (10.3.19) is orthogonal to the noise matrix $N$ as $N \to \infty$. The asymptotic condition $N \to \infty$ for the delay/Doppler estimation problem considered here is similar to the DOA estimation problem studied in [VON95], where the number of sensors was assumed to be large. Details on the properties of the ML estimator for this latter case can be found therein.

Next we present the CRB associated with the time-domain model (10.2.4). A similar CRB formula holds for the frequency domain model (10.2.8), but we focus on (10.2.4) here since it is the “exact” representation of our data. As a general rule, all estimators discussed in this chapter are based on the approximate frequency domain model but the data are generated by the “exact” time-domain equation and accordingly the CRB performance is evaluated for the latter model. The CRB is given by the inverse of the following Fisher Information Matrix (FIM) (see, e.g., [SM97, Kay93]):
\[
\text{FIM}_{k,p}(\xi) = \frac{2}{\sigma^2} \sum_{i=1}^{N} \text{Re} \left\{ \frac{\partial \mu^*_k}{\partial \xi_k} \frac{\partial \mu_k}{\partial \xi_p} \right\},
\]
where
\[
\xi = \left[ \begin{array}{cc} \omega^T & \tau^T \\ \text{vec} \left\{ \text{Re} \{ A \} \right\}^T & \text{vec} \left\{ \text{Im} \{ A \} \right\}^T \end{array} \right]^T
\]
\[
\mu_k = A^T \left[ \begin{array}{c} s(t - \tau_1)e^{j\omega_1 t} \\ \cdots \\ s(t - \tau_d)e^{j\omega_d t} \end{array} \right]^T,
\]
and $\text{Re} \{ A \}$ and $\text{Im} \{ A \}$ denote the real and imaginary parts of the spatial signature matrix $A$, and where $\text{vec}(\cdot)$ is the vectorization operator.
For the case where $\xi_i$ is a time delay or Doppler shift, the partial derivatives required to evaluate (10.3.21) can be written as follows:

$$\frac{\partial \mu_i}{\partial \tau_i} = A^T \begin{bmatrix} 0 & \ldots & 0 & -d(t - \tau_i)e^{j2\omega D_i t} & 0 & \ldots & 0 \end{bmatrix}^T,$$  \hspace{1cm} (10.3.24)

$$\frac{\partial \mu_i}{\partial \omega_{D_i}} = A^T \begin{bmatrix} 0 & \ldots & 0 & jts(t - \tau_i)e^{j2\omega D_i t} & 0 & \ldots & 0 \end{bmatrix}^T,$$  \hspace{1cm} (10.3.25)

where

$$d(t) = \frac{\partial s(t)}{\partial t}.$$  \hspace{1cm} (10.3.26)

The partial derivatives with respect to the real and imaginary parts of the $k, p$th element of $A$ can also be easily found.

In general, finding the ML estimates of $\tau$ and $\omega$ from (10.3.20) requires a multidimensional search over a $2d$-dimensional parameter space, which may be computationally burdensome. In the next section, we present a number of subspace-based methods that estimate the parameters of interest in a more computationally efficient manner. Some of these methods have accuracy comparable to that of the ML approach (and the CRB).

### 10.4 Subspace-Based Estimation Methods

In this section we describe algorithms for time delay and frequency/Doppler offset estimation based on Noise Subspace Fitting (NSF) [SS90, OVSN93], Signal Subspace Fitting (SSF) [SS90, VO91, OVSN93], MUSIC [Sch81] (see also [Hän99]) and ESPRIT [RK89]. It will be shown that due to the special structure of the signal manifold in the frequency domain, both NSF and SSF reduce to a $d$-dimensional search for the delay parameters. Of the two, SSF is expected to be more robust when the spatial signature matrix $A$ is nearly rank-deficient, or when the time delay differences are very small [OVSN93]. Both methods require initial estimates of the $\tau$ parameters, and for this purpose the MUSIC estimator and an ESPRIT-based estimator are derived. The MUSIC estimator requires $d$ one-dimensional searches, while the ESPRIT estimator, which ignores the Doppler shifts, does not require any search.
10.4.1 Noise Subspace Fitting

The NSF loss function for the problem at hand may be written as [SV93, SS90]

\[
V_{NSF}(\tau, \omega) = \text{tr} \left\{ Q^* (\tau, \omega) \hat{E}_n \hat{E}_n^* Q(\tau, \omega) \hat{U} \right\},
\]

(10.4.27)

where

\[
\hat{U} = Q^\dagger (\hat{\tau}, \hat{\omega}) \hat{E}_n W \hat{E}_n^* Q^\dagger (\hat{\tau}, \hat{\omega}),
\]

(10.4.28)

\(\hat{\tau}\) and \(\hat{\omega}\) are consistent (initial) estimates of \(\tau\) and \(\omega\), \(W\) is a diagonal weighting matrix, \(\hat{E}_s\) is the matrix whose columns are the left singular vectors corresponding to the \(d\) largest singular values of \(X\), and \(\hat{E}_n\) is a \(N \times (N - d)\) matrix whose columns are orthogonal to those of \(\hat{E}_s\). The choice of the matrix \(W\) depends on whether it is desired to approximate the so-called deterministic or stochastic ML solution (see [SS90, OVSN93] for details). In the simulations presented later, we use the stochastic ML weighting

\[
W = (\hat{\Lambda}_s - \hat{\sigma}^2 I)^2 \hat{\Lambda}_s^{-1},
\]

(10.4.29)

where \(\hat{\Lambda}_s\) is a diagonal matrix formed from the \(d\) largest squared singular values of \(X\), and \(\hat{\sigma}^2\) is a consistent estimate of the noise variance (obtained, for example, as the average of the \(m - d\) smallest squared singular values of \(X\)).

Introduce \(P = \hat{E}_n \hat{E}_n^*\), and write \(Q(\tau, \omega) = SV - DV\Phi\) by eliminating the explicit dependency on \(\tau\) and \(\omega\), so that (10.4.27) becomes

\[
\text{tr} \left\{ Q^* P Q \hat{U} \right\} = \text{tr} \left\{ V^* S^* P S V \hat{U} \right\} - \text{tr} \left\{ V^* S^* P D V \Phi \hat{U} \right\} - \text{tr} \left\{ \Phi^* V^* D^* P S V \hat{U} \right\} + \text{tr} \left\{ \Phi^* V^* D^* P D V \Phi \hat{U} \right\}.
\]

(10.4.30)

It is easily shown that

\[
\text{tr} \left\{ Y^* Z Y F \right\} = y^* \left( Z \odot F^T \right) y,
\]

(10.4.31)

where \(Z\) and \(F\) are arbitrary square matrices, and \(Y = \text{diag}(y)\) is a diagonal matrix. Using (10.4.30) and (10.4.31), the NSF criterion in (10.4.27) can be rewritten as

\[
V_{NSF}(\tau, \omega) = \eta^T (\omega) M(\tau) \eta(\omega)
\]

\[
= \eta^T (\omega) \text{Re} [M(\tau)] \eta(\omega)
\]

(10.4.32)
where

$$\eta(\omega) = \begin{bmatrix} \omega \\ e \end{bmatrix},$$

$$M(\tau) = \begin{bmatrix} (V^*D^*PDV) \odot \hat{U}^T & -(V^*D^*PSV) \odot \hat{U}^T \\ -(V^*S^*PDV) \odot \hat{U}^T & (V^*S^*PSV) \odot \hat{U}^T \end{bmatrix},$$

and $e = [1 \ \cdots \ \ 1]^T$ is $d \times 1$.

Since the NSF criterion is quadratic in $\omega$, the estimation of $\omega$ is separable from that of $\tau$. Setting $\partial V_{NSF}/\partial \omega = 0$ yields

$$\hat{\omega} = [\text{Re}(M_{11})]^{-1} \text{Re}(M_{12}) e,$$  \hspace{1cm} (10.4.33)

where the matrix $M(\tau)$ has been partitioned into $d \times d$ blocks:

$$M(\tau) = \begin{bmatrix} M_{11} & -M_{12} \\ -M_{21} & M_{22} \end{bmatrix}.$$

Substituting (10.4.33) into (10.4.32) leads to the following criterion for estimating $\tau$:

$$\hat{\tau} = \arg \min_{\tau} \eta^T(\hat{\omega})M(\tau)\eta(\hat{\omega})$$

$$= \arg \min_{\tau} e^T \left( \text{Re}(M_{22}) - \text{Re}(M_{12})^T [\text{Re}(M_{11})]^{-1} [\text{Re}(M_{12})] \right) e,$$  \hspace{1cm} (10.4.34)

which is the sum of the elements of the Schur complement of $\text{Re}(M_{11})$ in $\text{Re}(M(\tau))$.

It is worth mentioning that, since typically $N \gg d$, it is advantageous to compute $P$ as $P = I - \hat{E}_n \hat{E}_n^*$ rather than $P = \hat{E}_n \hat{E}_n^*$. The computation of $\hat{E}_n$ is cheaper than that of $\hat{E}_n$. Additionally, calculation of the submatrices of $M$ can be simplified if the required matrix products are performed in a certain order. For example, if we first compute the terms below in the parentheses

$$V^*D^*PDV = (V^*D^*) (DV) - \left(V^*D^* \hat{E}_s\right) \left(\hat{E}_s^*DV\right),$$  \hspace{1cm} (10.4.35)

the required computation is of order $O(Nd^2)$ instead of $O(N^2d)$.

The NSF algorithm is implemented by performing a $d$-dimensional search of the criterion in (10.4.34). As mentioned above, consistent initial estimates of $\tau$ and $\omega$ are required for computing the matrix $\hat{U}$ used in the NSF criterion. One way of obtaining $\hat{U}$ would be to first implement the
NSF algorithm with $\hat{\mathbf{U}} = \mathbf{I}$, and use the resulting estimates to form the optimal $\hat{\mathbf{U}}$. Setting $\mathbf{U} = \mathbf{I}$ is equivalent to using the MUSIC approach described later in this section. There are two drawbacks associated with the NSF algorithm: first, the algorithm is not always able to resolve closely spaced components in $\tau$, and second, the algorithm’s performance may deteriorate when the rows of $\mathbf{A}$ are linearly dependent, which can occur when either $d > m$, or two arrivals with different delays share the same spatial signature. The SSF algorithm presented in the next section overcomes these two drawbacks.

### 10.4.2 Signal Subspace Fitting

The SSF estimates of the delays and frequency/Doppler offsets can be found by minimizing [SS90, VO91, OVSN93]

$$V_{SSF}(\tau, \omega) = \text{tr} \left\{ \Pi_{\hat{Q}} \hat{E}_s \mathbf{W} \hat{E}_s^* \right\},$$

(10.4.36)

where the diagonal weighting $\mathbf{W}$ is as defined in (10.4.29). As shown below, the Doppler parameters can also be explicitly estimated using SSF, but only for the case where $d < N/2$, which is not a serious restriction in most cases (see the discussion on identifiability in Section 10.2.3).

Define

$$\mathbf{C} = \begin{bmatrix} \mathbf{S} \mathbf{V} & -\mathbf{D} \mathbf{V} \end{bmatrix},$$

(10.4.37)

and suppose that $d < N/2$ and $\mathbf{C}$ has full column rank. Let $\mathbf{B}$ be an $N \times (N - 2d)$ matrix that spans the space orthogonal to the columns of $\mathbf{C}$ ($\mathbf{B}$ is used as a dummy variable in what follows), and define the $(N - d) \times N$ matrix $\mathbf{H}^*$ as

$$\mathbf{H}^* = \begin{bmatrix} \mathbf{B}^* \\ \mathbf{\Phi}^* & -\mathbf{I} \end{bmatrix} \mathbf{C}^\dagger.$$

(10.4.38)

Observe that

$$\mathbf{H}^* \mathbf{Q} = \mathbf{H}^* \mathbf{C} \begin{bmatrix} \mathbf{I} \\ \mathbf{\Phi} \end{bmatrix} = 0,$$

and that

$$\mathbf{H}^* \mathbf{H} = \begin{bmatrix} \mathbf{B}^* \mathbf{B} & 0 \\ 0 & \mathbf{\Phi}^* \end{bmatrix} - \mathbf{I} \mathbf{C}^\dagger \mathbf{C} \mathbf{C}^\dagger,$$

is nonsingular. Thus, the columns of $\mathbf{H}$ span the nullspace of $\mathbf{Q}^*$, and the projection matrix $\Pi_{\hat{Q}}^*$ can be written as

$$\Pi_{\hat{Q}}^* = \Pi_{\mathbf{H}} = \mathbf{H} \mathbf{H}^\dagger$$

$$= \mathbf{I} - \mathbf{C} \mathbf{C}^\dagger + \mathbf{C}^\dagger \mathbf{\Phi}^* \mathbf{\Phi}^* \mathbf{\Phi} \mathbf{\Phi} - \mathbf{I} \mathbf{C}^\dagger \mathbf{C}^\dagger,$$

(10.4.39)
where
\[
T^{-1} = \begin{bmatrix} \Phi & -I \end{bmatrix} (C^* C)^{-1} \begin{bmatrix} \Phi^* \\ -I \end{bmatrix}.
\] (10.4.40)

To make the SSF criterion quadratic in \( \Phi \), we assume that \( T \) is calculated using some initial consistent estimate of the parameters. With the resulting matrix \( \hat{T} \), we rewrite (10.4.36) approximately as
\[
V_{SSF} = \text{tr} \left\{ \left( I - CC^\dagger \right) \hat{E}_s \hat{W} \hat{E}_s^\dagger \right\} + \text{tr} \left\{ \Psi \right\},
\] (10.4.41)

where
\[
\Psi = \Phi \Gamma_{11} \Phi^* - \Gamma_{12} \Phi^* - \Phi \Gamma_{12} + \Gamma_{22}
\] (10.4.42)

and \( \Gamma_{ij} \) are the \( d \times d \) blocks of the matrix
\[
\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \triangleq C^\dagger \hat{E}_s \hat{W} \hat{E}_s \hat{C}^\dagger.
\] (10.4.43)

Replacing \( T \) by the consistent estimate \( \hat{T} \) does not affect the asymptotic performance of the algorithm since
\[
\begin{bmatrix} \Phi & -I \end{bmatrix} C^\dagger \hat{E}_s = o(1)
\]
at the true values of \( \tau \) and \( \omega \).

Introducing \( \alpha \) as the vector formed from the diagonal elements of \( \Gamma_{12} \hat{T} \), defining
\[
\Omega = \hat{T} \otimes \Gamma_{11}^T
\] (10.4.44)

\[
\rho = \text{tr} \left\{ \Gamma_{22} \hat{T} \right\},
\] (10.4.45)

and using (10.4.31) once again, the part of the cost function that depends on the Doppler shifts can be written as
\[
\text{tr} \left\{ \Psi \right\} = \omega^* \Omega \omega - \alpha^* \omega - \omega^* \alpha + \rho
\]
\[
= \omega^T [\text{Re}(\Omega)] \omega - 2 \omega^T \text{Re}(\alpha) + \rho.
\] (10.4.46)

Minimization of (10.4.46) with respect to \( \omega \) yields the estimate
\[
\hat{\omega} = [\text{Re}(\Omega)]^{-1} \text{Re}(\alpha).
\] (10.4.47)

Inserting (10.4.47) into (10.4.41) leads to the following estimate of \( \tau \):
\[
\hat{\tau} = \arg \min_{\tau} \left[ \rho - [\text{Re}(\alpha)]^T [\text{Re}(\Omega)]^{-1} [\text{Re}(\alpha)] - \text{tr} \left\{ C C^\dagger \hat{E}_s \hat{W} \hat{E}_s \right\} \right].
\] (10.4.48)
Note that the computation required to evaluate the SSF criterion can be significantly simplified by performing the trace calculation in (10.4.48) as
\[
\text{tr} \left\{ C C^T \mathbf{\hat{E}}_s W^T \mathbf{E}_s^* \right\} = \text{tr} \left\{ \left( C C^T \right) W \left( \mathbf{E}_s^* C \right) \right\}.
\]

(10.4.49)

The SSF algorithm is implemented by performing the \( d \)-dimensional search in (10.4.48). As with NSF, the SSF method requires consistent initial estimates of both \( \tau \) and \( \omega \) to form \( \mathbf{T} \), which is then used in calculating \( \Omega \) and \( \alpha \). Such estimates can be obtained using either the MUSIC or the ESPRIT approaches presented in the following subsections, or by an initial application of SSF with \( \mathbf{T} = \mathbf{I} \). It will be seen in Section 10.5 that, as predicted, SSF outperforms NSF in cases where elements of \( \tau \) are closely spaced or when \( \text{rank} \left( \mathbf{A} \right) < d \) or nearly so. However, the algorithms perform almost identically at high SNRs.

10.4.3 MUSIC

In the standard MUSIC algorithm [Sch81] for DOA estimation, the DOAs are determined to be the \( d \) values of \( \theta \) that make \( \mathbf{a}(\theta) \) nearly orthogonal to \( \mathbf{E}_n \), according to the following measure:
\[
\mathbf{V}_M(\theta) = \frac{\mathbf{a}^T(\theta) \mathbf{E}_n \mathbf{\hat{E}}_n^* \mathbf{a}(\theta)}{\mathbf{a}^T(\theta) \mathbf{a}(\theta)}.
\]

(10.4.50)

In the delay and Doppler estimation problem, assuming that \( \text{rank} \left( \mathbf{A} \right) = d \), we replace \( \mathbf{a}(\theta) \) with the signal’s frequency signature
\[
\mathbf{q}(\tau, \omega_D) = \mathbf{Sv}(\tau) - \omega_D \mathbf{Dv}(\tau)
\]
\[
\overset{\Delta}{=} \mathbf{G}(\tau) \mathbf{g}(\omega_D),
\]

(10.4.51)

where \( \mathbf{g}(\omega) = \left[ \begin{array}{c} 1 \\ \omega_D \end{array} \right] \mathbf{T} \), and
\[
\mathbf{G}(\tau) = \left[ \begin{array}{cc} \mathbf{Sv}(\tau) & -\mathbf{Dv}(\tau) \end{array} \right].
\]

For this case, the MUSIC loss function becomes
\[
\mathbf{V}_M(\tau, \omega) = \frac{\mathbf{g}^T(\omega) \left[ \text{Re} \left( \mathbf{G}^T(\tau) \mathbf{\hat{E}}_n \mathbf{\hat{E}}_n^* \mathbf{G}(\tau) \right) \right] \mathbf{g}(\omega)}{\mathbf{g}^T(\omega) \left[ \text{Re} \left( \mathbf{G}^T(\tau) \mathbf{G}(\tau) \right) \right] \mathbf{g}(\omega)},
\]

(10.4.52)

since \( \mathbf{g}(\omega) \) is real-valued. The MUSIC criterion in (10.4.52) is seen to be a ratio of quadratic forms in \( \mathbf{g}(\omega) \), and thus minimizing \( \mathbf{V}_M(\tau, \omega) \) with respect to \( \mathbf{g}(\omega) \) is equivalent to finding, as a function of \( \tau \), the minimum
generalized eigenvalue and associated eigenvector of the following $2 \times 2$ matrices:

$$
\text{Re} \left( G^*(\tau) \tilde{E}_n \tilde{G}^*(\tau) \right) \gamma_{\min} = \lambda_{\min} \text{Re} \left( G^*(\tau) G(\tau) \right) \gamma_{\min}.
$$

(10.4.53)

As in the algorithms of [Sch81, FP83], the time delays can be found by viewing $\lambda_{\min}$ as a function of $\tau$, and searching for the $d$ deepest minima of $\lambda_{\min}(\tau)$. The corresponding frequency offsets are then calculated using the generalized eigenvector associated with $\lambda_{\min}(\tau)$:

$$
\hat{\omega}_D = \frac{\gamma_{\min,2}(\tilde{\tau}_k)}{\gamma_{\min,1}(\tilde{\tau}_k)},
$$

(10.4.54)

where $\gamma_{\min,i}$ is element $i$ of $\gamma_{\min}$.

Note that the matrix $G^*(\tau)G(\tau)$ is actually independent of $\tau$:

$$
G^*(\tau)G(\tau) = \begin{bmatrix}
\|s\|^2 & -s^*d \\
-d^*s & \|d\|^2
\end{bmatrix},
$$

a fact which simplifies evaluation of the above generalized MUSIC criterion. This observation also implies that the signal frequency sample vector $s$ and the associated gradient vector $d$ must be linearly independent for the algorithm to work.

### 10.4.4 ESPRIT

In [Swi98], an algorithm based on ESPRIT was presented for estimating time delays in cases where the frequency/Doppler offset is zero. Our empirical results indicate that this approach still gives reasonable time delay estimates even when the frequency offset is non-zero but small. The fact that the algorithm yields the desired estimates in closed form (i.e., without a search) makes it an attractive alternative for initializing the SSF and NSF searches. The ESPRIT approach of [Swi98] is briefly described below. Note that the algorithm is only applicable in cases where the rows of $A$ are linearly independent (i.e., when $\text{rank}(A) = d < m$).

Consider the frequency domain representation of (10.2.8) for the case with no frequency/Doppler offset:

$$
Q = S \mathbf{V}(\tau).
$$

(10.4.55)

Let $V_1$ be the $(N - \delta) \times d$ matrix made from the first $N - \delta$ rows of $\mathbf{V}(\tau)$, and let $V_2$ be constructed similarly from the last $N - \delta$ rows. If the data
is evaluated at the DFT frequencies \( \omega_k = 2\pi k/N, k = 1, \cdots, N \), then \( V(\tau) \) is Vandermonde, and its submatrices are related as follows:

\[
V_2 = V_1 \Upsilon
\]

(10.4.56)

where

\[
\Upsilon = \begin{bmatrix}
e^{-j2\pi \delta \tau_1 /N} & \cdots & 0 \\
\vdots & \ddots & \vdots \\
e^{-j2\pi \delta \tau_d /N} & \cdots & e^{-j2\pi \delta \tau_1 /N}
\end{bmatrix}.
\]

(10.4.57)

Let \( E_s \) denote the noise-free left singular vectors of \( X \), and similarly define, for \( E_s \) and \( S \), the submatrices \( E_1, E_2, S_1, S_2 \). Since, under the assumption that \( \text{rank}(A) = d \), the columns of \( E_s \) and \( Q = SV(\tau) \) span the same subspace, we have

\[
E_1 = S_1 V_1 T_e
\]

(10.4.58)

\[
E_2 = S_2 V_2 T_e
\]

(10.4.59)

for some \( d \times d \) matrix \( T_e \). Combining (10.4.56) – (10.4.59) and eliminating \( V_1 \) and \( V_2 \) leads to

\[
S_1 E_2 = S_2 S_1 V_1 \Upsilon T_e = S_2 E_1 Z,
\]

(10.4.60)

where

\[
Z = T_e^{-1} \Upsilon T_e.
\]

Provided that \( \delta \) is chosen small enough so that

\[
\frac{N}{\delta} > \max_k \tau_k,
\]

(10.4.61)

there is a one-to-one relationship between the delays and the eigenvalues of \( Z \). With noisy data, \( Z \) must be estimated using \( E_s \) and its submatrices \( E_1 \) and \( E_2 \). A simple least squares estimate of \( Z \) is given by

\[
\hat{Z} = (S_2 E_1)^{\dagger} S_1 E_2.
\]

If \( \hat{\lambda}_k \) denotes the \( k \)th eigenvalue of \( \hat{Z} \), the time delay estimates can be found from

\[
\hat{\tau}_k = \frac{-N \arg \hat{\lambda}_k}{2\pi \delta}.
\]

(10.4.62)

As shown in [RGO+89, GR90, Swi98], the variance of the ESPRIT estimates can be reduced by choosing the overlap factor \( \delta \) to be as large as possible, while still satisfying (10.4.61). In the simulations presented below, we use \( \delta = 1 \) for simplicity.

While the above ESPRIT approach yields estimates of the time delays, denoted by \( \hat{\tau}_e \), both NSF and SSF also require initial Doppler estimates to
form the matrices $\hat{U}$ and $\hat{T}$. For NSF, an initial Doppler estimate can be obtained by substituting $\hat{\tau}_e$ into (10.4.33) with either $\hat{U} = I$, or $\hat{U}$ evaluated using $\hat{\tau} = \hat{\tau}_e$ and $\hat{\omega} = 0$. Alternatively, for SSF, equation (10.4.47) can be used, either with $\hat{T} = I$, or $\hat{T}$ evaluated using $\hat{\tau} = \hat{\tau}_e$ and $\hat{\omega} = 0$. A third approach would be to use MUSIC and the generalized eigenvector in (10.4.53) for every element of $\hat{\tau}_e$, and then solve for the associated Doppler frequency using (10.4.54). All of these methods performed equally well in the numerical studies we have conducted. In the simulation results presented in Section 10.5, the initial Doppler estimates for NSF and SSF were obtained using (10.4.33) with $\hat{U} = I$ and (10.4.47) with $\hat{T} = I$, respectively.

An ESPRIT solution similar to the one presented above can also be arrived at under a slightly different set of assumptions, without ignoring the Doppler component of the model. Using the notation developed above, but with the Doppler parameters included, the partitioned matrices of singular vectors will satisfy

$$E_1 = (S_1 V_1 - D_1 V_1 \Phi) T_e$$  \hspace{1cm} (10.4.63)

$$E_2 = (S_2 V_2 - D_2 V_2 \Phi) T_e ,$$  \hspace{1cm} (10.4.64)

where $D_1$ and $D_2$ are formed from $D$ in the same way that $S_1$ and $S_2$ are formed from $S$. Using (10.4.66), equation (10.4.64) may be rewritten as

$$E_2 = (S_2 V_1 - D_2 V_1 \Phi) \Upsilon T_e ,$$  \hspace{1cm} (10.4.65)

since the diagonal matrices $\Phi$ and $\Upsilon$ commute. Now, if it is assumed that the signal spectrum is smooth enough so that $S_1 \simeq S_2$ and $D_1 \simeq D_2$, then

$$E_2 \simeq (S_1 V_1 - D_1 V_1 \Phi) \Upsilon T_e$$  \hspace{1cm} (10.4.66)

$$= E_1 T_e^{-1} \Upsilon T_e$$  \hspace{1cm} (10.4.67)

$$= E_1 Z ,$$  \hspace{1cm} (10.4.68)

and once again the time delays can be found from the eigenvalues of $Z = E_1 E_2$, which is equal to the matrix $Z$ defined above for the case where $S_1 = S_2$. We note here that the smoothness condition $S_1 \simeq S_2$ was also used in [BD89] to find a simple ESPRIT time delay solution for the case of zero Doppler.
10.5 Numerical examples

10.5.1 Performance versus SNR

We begin by studying the performance of the algorithms presented above as the signal-to-noise ratio (SNR) varies. Simulation data was generated using (10.2.1) for two multipath signals \( d = 2 \) with time delays \( \tau = [ 0.5 \ 3 ]^T \), and Doppler shifts \( \omega = 2\pi [ -0.02 \ 0.01 ]^T \). The data was corrupted by spatially and temporally white circular Gaussian noise with zero mean and standard deviation \( \sigma \). The two columns of the signature matrix, \( \mathbf{A} \), were given by the array response of a 5-element, half-wavelength spaced, uniform linear array (ULA), to far-field sources with DOAs of 0° and 20°. The signal sequence was chosen to be a unit power raised cosine function, a commonly occurring signal both in communications and radar applications:

\[
s(t) = \frac{\text{sinc}(t/T)}{1 - (t/T)^2}.
\]

For the simulations presented below, \( T = 5 \), and \( N = 101 \) samples are assumed to be taken from the array. The root mean squared error (rMSE) of the time delay and Doppler estimates were calculated for each of the algorithms based on 500 Monte Carlo trials for various SNR values. The results are plotted in Figure 10.1, together with the appropriate CRB. The ESPRIT time delay estimates (dotted line) were, as described in section 10.4.4, used as the initial values for the NSF (dashed line) and the SSF (upper solid line) searches. Furthermore, the ESPRIT time delay estimates were also used to initialize the local search for the MUSIC (dash-dotted line) time delay estimates. As can be seen from the figures, all of the algorithms except ESPRIT achieve the CRB at about SNR = 4 dB. The excess error for the ESPRIT algorithm is of course due to the fact that it assumes the Doppler is zero when estimating the time delays; thus it yields biased time delay estimates. The NSF and MUSIC algorithms failed to resolve the two signals in 10-20% of the trials below SNR=4 dB (in which cases the two estimates were identical within numerical accuracy); when such a failure occurred, the single parameter value was assigned to be the estimate for both arrivals to compute the rMSE. It is interesting to note that the NSF Doppler estimates are actually below the CRB for low SNRs. This is due to the fact that the NSF Doppler estimates become biased in this region, and approach zero as SNR → 0. The excess bias for NSF is evident in Figure 10.2, which plots the ratio of the bias and standard deviation of the Doppler estimate for the first arrival.
Figure 10.1: The rMSE of the proposed estimators as a function of the SNR compared with the corresponding CRB. (a) time delay, $\tau_1$. (b) time delay, $\tau_2$. (c) Doppler shift, $\omega_1$. (d) Doppler shift, $\omega_2$. 
10.5.2 Closely spaced sources

Next, we study how the performance of the estimators depends on the assumption of a full rank spatial signature matrix. The data studied is as described above, but with SNR = 15 dB and DOAs [0°, θ], where the DOA of the second arrival, θ, is varied from 0° to 25°. Figures 10.3(a) and 10.3(b) compare the estimation errors for the time delay estimates, and as can be seen from the figures, the NSF, the SSF and the MUSIC estimates achieve the CRB when the angular difference is above 15°. Furthermore, it is seen that all of the algorithms degrade significantly for angular differences lower than 5°. A nearly rank deficient A matrix for our model corresponds to the coherent signal case in DOA estimation. Thus we expect that MUSIC, NSF, and ESPRIT will all have poor performance in this case, and this is evident in the plots. SSF also degrades here due to the fact that it uses ESPRIT for initialization. Figures 10.3(c) and 10.3(d) compare the rMSE for the Doppler shift estimates with the corresponding CRB. Here, the NSF and the MUSIC estimates are found to have a somewhat lower rMSE than the SSF estimates.
Figure 10.3: The rMSE of the proposed estimators as a function of the DOA difference compared with the corresponding CRB. (a) time delay, $\tau_1$. (b) time delay, $\tau_2$. (c) Doppler shift, $\omega_1$. (d) Doppler shift, $\omega_2$. 
10.5.3 Large Doppler shifts

In this case, we examine the robustness of the algorithms to the assumption that the Doppler shifts are small. We remind the reader that the data are generated with (10.2.4) whereas all estimators use the model (10.2.8) obtained under the assumption of small Doppler offsets. The data studied is as described in Section 10.5.1, but with SNR = 15 dB and with \( \omega = \begin{bmatrix} -0.02 & \rho \end{bmatrix} \), where the Doppler shift of the second arrival, \( \rho \), is varied from 0 to 0.1. Figures 10.4(a) and 10.4(b) compare the estimation errors for the time delay estimates with the corresponding CRB. Figures 10.4(c) and 10.4(c) compare the Doppler shift estimation errors, and as can be seen from the figures, all the algorithms prove to be very robust to relatively large Doppler shifts.

10.5.4 Resolving closely-spaced arrivals

In our final example, we investigate the ability of the algorithms to resolve closely-spaced arrivals. The case studied is as described previously in Section 10.5.1, but with SNR = 15 dB and with time delays \( \tau = \begin{bmatrix} 0.5 & 0.5 + \Delta \end{bmatrix} \), where the time delay difference \( \Delta \) is varied from 1.5 to 4. Figures 10.5(a) and 10.5(b) compare the estimation errors for the time delays. As can be seen from the figures, the NSF, the SSF and the MUSIC estimates achieve the CRB for a time delay difference larger than 2. It was found that NSF was unable to resolve the two arrivals for \( \Delta < 2 \), while for MUSIC the threshold was slightly better, as it lost resolution for \( \Delta < 1.5 \). Figures 10.5(c) and 10.5(d) compare the estimation errors for the Doppler shifts. It is seen that for this case the algorithms performance are not significantly affected by variations in the time delay difference.

10.6 Conclusions

In this chapter, we have developed a data model for the time delay and Doppler shift estimation problem. We have illustrated the connection between the presented model and the polarization and angular spread models used in the DOA estimation case. Using the similarity between the models, we have developed general subspace-based time delay and Doppler estimators. These estimators only require a search for the time delay estimates since it is found that, to first order, the subspace depends linearly on the Doppler parameters for small Doppler shifts. The performance of the algorithms was examined via several simulations, and it was found that the algorithms perform very well, even for large Doppler shifts well beyond
those typically encountered in practice. For the cases considered, the SSF algorithm performed the best, and when it was able to resolve the sources MUSIC had similar or in some cases even slightly better performance. On the other hand, the NSF algorithm had more trouble resolving arrivals for difficult cases. Though biased, ESPRIT was found to be an effective method for initializing the search-based methods.
Figure 10.4: The rMSE of the proposed estimators as a function of the Doppler shift difference compared with the corresponding CRB. (a) time delay, $\tau_1$. (b) time delay, $\tau_2$. (c) Doppler shift, $\omega_1$. (d) Doppler shift, $\omega_2$. 
Figure 10.5: The rMSE of the proposed estimators as a function of the time delay difference compared with the corresponding CRB. (a) time delay, $\tau_1$. (b) time delay, $\tau_2$. (c) Doppler shift, $\omega_1$. (d) Doppler shift, $\omega_2$. 
Bibliography


[Bro92] T. B. Bones. “On the Performance Advantage of Multi-

Stanford University, 1975.

in spectrum analysis and array processing*, volume II, pages 1–


[Cap71] J. Capon. “Correction to ‘Probability Distributions for Estima-
tors of the Frequency-Wavenumber Spectrum’”. *Proc. of the

S. Haykin, ed. Springer-Verlag, New York, USA, 1983.

Estimators of the Frequency-Wavenumber Spectrum”. *Proc. of the


[CRP80] Y. T. Chan, J. M. Riley, and J. B. Plant. “A Parameter Estima-


[CH99] V. C. Chen and Ling H. “Joint Time-Frequency Analysis for
Radar Signal and Image Processing”. *IEEE Signal Processing

Doppler-Shifted Gaussian Random Processes”. In *Proc. 8th
Workshop on Stat. Sig. and Array Proc.*, pages 28–31, Corfu,
Greece, June 1996.


<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title/Description</th>
<th>Journal/Source</th>
</tr>
</thead>
</table>


I. Newton. *Phil. Trans.*, vol. VI, 1671. On page 3076: ”Comparing the length of this coloured spectrum with its breadth...”. 


