

Spectral Analysis of Nonuniformly
Sampled Data and Applications

Spectral Analysis of Nonuniformly Sampled Data and Applications

Prabhu Babu



UPPSALA
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in Polacksbacken, Hus 2, room: P2347, Department of Information Technology, Uppsala, Friday, October 19, 2012 at 13:00 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract

Babu, P. 2012. Spectral Analysis of Nonuniformly Sampled Data and Applications. Institutionen för informationsteknologi. 220 pp. Uppsala. ISBN 978-91-506-2300-0.

Signal acquisition, signal reconstruction and analysis of spectrum of the signal are the three most important steps in signal processing and they are found in almost all of the modern day hardware. In most of the signal processing hardware, the signal of interest is sampled at uniform intervals satisfying some conditions like Nyquist rate. However, in some cases the privilege of having uniformly sampled data is lost due to some constraints on the hardware resources. In this thesis an important problem of signal reconstruction and spectral analysis from nonuniformly sampled data is addressed and a variety of methods are presented. The proposed methods are tested via numerical experiments on both artificial and real-life data sets.

The thesis starts with a brief review of methods available in the literature for signal reconstruction and spectral analysis from non uniformly sampled data. The methods discussed in the thesis are classified into two broad categories - dense and sparse methods, the classification is based on the kind of spectra for which they are applicable. Under dense spectral methods the main contribution of the thesis is a non-parametric approach named LIMES, which recovers the smooth spectrum from non uniformly sampled data. Apart from recovering the spectrum, LIMES also gives an estimate of the covariance matrix. Under sparse methods the two main contributions are methods named SPICE and LIKES - both of them are user parameter free sparse estimation methods applicable for line spectral estimation. The other important contributions are extensions of SPICE and LIKES to multivariate time series and array processing models, and a solution to the grid selection problem in sparse estimation of spectral-line parameters.

The third and final part of the thesis contains applications of the methods discussed in the thesis to the problem of radial velocity data analysis for exoplanet detection. Apart from the exoplanet application, an application based on Sudoku, which is related to sparse parameter estimation, is also discussed.

Keywords: Spectral analysis, array processing, nonuniform sampling, sparse parameter estimation, direction of arrival (DOA) estimation, covariance fitting, sinusoidal parameter estimation, maximum-likelihood, non-parametric approach, exoplanet detection, radial velocity technique, Sudoku.

Prabhu Babu, Uppsala University, Department of Information Technology, Division of Systems and Control, Box 337, SE-751 05 Uppsala, Sweden.

© Prabhu Babu 2012

ISBN 978-91-506-2300-0

urn:nbn:se:uu:diva-180391 (<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-180391>)

*To the souls of over 40000 people (including women and children) who
were killed in Mullivaikal, Srilanka on 18 May 2009.*

Contents

1	Introduction	11
1.1	Thesis outline	11
1.2	Topics for future research	17
Part I: Dense methods		19
2	Methods for signal reconstruction and spectral analysis from nonuniformly sampled data - a review	21
2.1	Introduction	21
2.2	Notations and Preliminaries	24
2.2.1	Signal models	24
2.2.2	Sampling patterns	25
2.2.3	Resampling and interpolation techniques	27
2.2.4	Least Squares, Maximum Likelihood	28
2.3	Signal Reconstruction Methods	29
2.3.1	Nyquist frequency (Ω_{nus})	31
2.3.2	Frequency resolution ($\Delta\Omega$)	31
2.3.3	Resampling time (t_r)	32
2.4	Spectral Analysis Methods	32
2.4.1	Methods based on least squares	33
2.4.2	Methods based on interpolation	35
2.4.3	Methods based on slotted resampling	37
2.4.4	Methods based on continuous time models	42
2.4.5	Classification and summary	43
2.4.6	Performance on real life data sets	44
2.5	Conclusions	49
3	A nonparametric approach to estimation of smooth spectra from nonuniformly sampled data	50
3.1	Introduction	50
3.2	Problem formulation and DAM	51
3.3	Non-parametric model of R	54
3.4	ML approach and the CRB	56
3.5	LIMES - the basic idea	58
3.6	LIMES - the algorithm	59
3.7	Numerical illustrations and concluding remarks	62
3.8	Evaluation of the integrals in (3.18) and (3.19)	72

3.9	Algebraic proof of (3.57)	74
3.10	Proof of (3.63) and (3.64)	74
4	A nonparametric approach to estimation of smooth spectra from nonuniformly sampled data - enhanced version	76
4.1	Introduction and brief review of LIMES	76
4.2	Averaged LIMES for very large data sets	77
4.3	Smoothed LIMES for small/medium data sets	78
4.4	Concluding remarks	80
	Part II: Sparse methods	83
5	Spectral analysis of nonuniformly sampled data via a sparse parameter estimation approach: SPICE	85
5.1	Introduction and problem formulation	85
5.2	The competing methods : IAA and SLIM	88
5.2.1	IAA	89
5.2.2	SLIM	89
5.3	The proposed method : SPICE	90
5.3.1	Convexity of the problem	91
5.3.2	Derivation of SPICE	91
5.3.3	Some theoretical properties	95
5.3.4	Some extensions	97
5.4	Application to spectral analysis and numerical performance study	99
5.5	Concluding remarks	103
5.6	Proof of (5.26) and (5.27)	109
5.7	On the minimizers of the SPICE criterion in the noise-free case	109
6	On the SPICE approach for direction of arrival estimation problem ..	111
6.1	Introduction and preliminaries	111
6.2	SPICE estimation criterion	113
6.3	SPICE updating formulas	117
6.3.1	The case of different $\{\sigma_k\}$	118
6.3.2	The case of identical $\{\sigma_k\}$	119
6.4	SOCP formulation of SPICE	120
6.5	Numerical illustrations and concluding remarks	123
6.5.1	Fixed sources : DOA estimation	123
6.5.2	Mobile sources : DOA tracking	124
6.6	Some SDP and SOCP formulations	126
6.7	Equivalence of (6.18) and (6.20)	130
7	On two user parameter free sparse parameter estimation approaches : SPICE and LIKES	132

7.1	Introduction and problem formulation	132
7.2	SPICE	133
7.2.1	SOCP-based solver	136
7.2.2	CA-based solver	138
7.3	LIKES	139
7.4	Numerical illustrations and concluding remarks	142
7.4.1	Spectral analysis example	142
7.4.2	Range-Doppler imaging example	144
7.5	Concavity proof	150
7.6	Non-convexity proof	150
7.7	Connection between SPICE and the square-root Lasso	151
8	On Grid selection problems and their solutions for sparse estimation of spectral lines	153
8.1	Introduction and problem formulation	153
8.2	Grid selection : preliminary ideas	155
8.2.1	Practical guideline	155
8.2.2	Theoretical guideline	155
8.2.3	Numerical example	157
8.3	Grid selection : refined ideas	160
8.3.1	LIKES	160
8.3.2	ReLIKES	161
8.3.3	SeLIKES	162
8.4	Numerical illustrations and concluding remarks	163
9	Sparse spectral-line estimation for non-uniformly sampled multivariate time series	165
9.1	Data model and problem formulation	165
9.2	SPICE, LIKES and MSBL	168
9.2.1	SPICE	168
9.2.2	LIKES	169
9.2.3	MSBL	171
9.3	Numerical simulations and concluding remarks	172
9.3.1	Statistical performance	172
9.3.2	Complexity and convergence rate	173
9.3.3	Concluding remarks	175
	Part III: Applications	177
10	Exoplanet detection : Analysis of radial velocity data	179
10.1	Introduction	179
10.2	Data Model	180
10.3	IAA based methods	182
10.3.1	IAA	182
10.3.2	Regularized IAA methods	182

10.3.3	Computational aspects and Range-selective IAA	184
10.4	Refining the estimates and statistical significance testing: RELAX and GLRT	186
10.5	Real life radial velocity data sets	189
10.5.1	HD 63454	191
10.5.2	HD 208487	191
10.5.3	GJ 876	193
10.6	Conclusions	195
11	A combined linear programming-maximum likelihood approach to radial velocity data analysis	197
11.1	Introduction and the data model	197
11.2	SPICE	199
11.3	Enhanced parameter estimation and significance testing : RELAX and GLRT	200
11.4	Real-life radial-velocity data processing	202
11.5	Conclusions	202
12	Linear Systems, Sparse Solutions, and Sudoku	205
12.1	Introduction	205
12.2	Sudoku ruleset as a linear system	205
12.3	l_0 norm minimization	208
12.4	l_1 norm minimization	208
12.5	Conclusions	211
	Acknowledgements	212
	References	213

1. Introduction

Spectral analysis of nonuniformly sampled data is an old and well known area of research. The nonuniformity in the data is generally due to reasons like unavailability of samples at specific instants of uniformly sampled data (commonly called missing data problem), random sampling device following a specific distribution, sampling device with arbitrary jitter around the regular uniform grid, sampling device following a periodic sampling pattern but arbitrary within each period, and sampling device with completely arbitrary sampling scheme. The problem of spectral analysis of nonuniformly sampled data is well motivated both by its theoretical significance and by its wide spread application in fields like astronomy, seismology, paleoclimatology, genetics and laser Doppler velocimetry. In this thesis we will discuss various methods that we have developed for spectral analysis of nonuniformly sampled data and include some practical applications of our methods. The chapters in this thesis are broadly classified into three parts:

Part 1 : Dense methods - This part deals with methods for recovering dense spectra from nonuniformly sampled data. Chapters 2 , 3 and 4 are included under this part.

Part 2 : Sparse methods - This part deals with methods for recovering sparse spectra from nonuniformly sampled data. Chapters 5, 6, 7, 8 and 9 are included under this part.

Part 3 : Applications - This part deals with applications of the methods discussed in part 1 and part 2. Chapters 10, 11 and 12 are included under this part.

In the following we briefly outline the contents of the chapters in the thesis.

1.1 Thesis outline

Chapter 2

In this chapter a brief review of different methods available in the literature for signal reconstruction and spectral analysis of nonuniformly sampled data is presented. The assumptions behind different methods for exact signal reconstruction are discussed and their computational complexities are compared. Methods for spectral analysis are classified under three broad categories and discussed. Finally, the performance of the methods are compared via numerical simulations on artificial and real-life data sets. The material in this chapter

is based on:

- P. Babu and P. Stoica, Spectral analysis of nonuniformly sampled data - a review, Digital Signal Processing, vol. 20, no. 2, pp. 359-378, 2010.

Chapter 3

In this chapter a maximum-likelihood method (named LIMES - likelihood-based method for estimation of spectra) for the non-parametric estimation of smooth spectra from nonuniformly sampled observations is discussed. An estimate of the data covariance matrix is obtained as a byproduct of the LIMES method. The performance of LIMES is evaluated via numerical simulations and a comparison with conventional methods like the Daniell method is discussed. The material in this chapter is based on:

- P. Stoica and P. Babu, Maximum-Likelihood nonparametric estimation of smoothed spectra from irregularly sampled data, IEEE Transactions on Signal Processing, vol. 60, pp. 5746-5758, 2012.

Chapter 4

In this chapter two problems of LIMES method, namely it is computationally intensive for large data sets and its high variance for data sets with short lengths, are addressed. The solutions proposed for the above mentioned problems are simple and their performance on artificial data sets are discussed. The material in this chapter is based on:

- P. Stoica and P. Babu, On the LIMES approach to the spectral analysis of irregularly sampled data, Electronics Letters, vol. 48, no. 4, 2012.

Chapter 5

In this chapter a new semiparametric/sparse method called SPICE (a semi-parametric/sparse iterative covariance-based estimation method) is derived and discussed. The merits of SPICE, namely it is free of any hyperparameters, it is computationally efficient, and it is globally converging are discussed. The statistical performance of SPICE is illustrated by means of a line-spectrum estimation study for irregularly-sampled data. The material in this chapter is based on:

- P. Stoica, P. Babu, and J. Li, New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sam-

pled data, IEEE Transactions on Signal Processing, vol. 59, pp.35-47, 2011.

Chapter 6

In this chapter the SPICE algorithm is applied to the direction of arrival estimation problem in array processing. The SPICE approach discussed in this chapter is obtained by the minimization of a covariance matrix fitting criterion which is particularly useful in many-snapshot cases but can be used even in single-snapshot situations as well. Numerical simulations based on uniform and nonuniform arrays are included to illustrate the performance of SPICE. The material in this chapter is based on:

- P. Stoica, P. Babu, and J. Li, SPICE: a novel covariance-based sparse estimation method for array processing, IEEE Transactions on Signal Processing, vo1.59, pp. 629-638, 2011.
- P. Stoica, P. Babu and J. Li, A sparse covariance based method for direction of arrival estimation, 36th International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, 2011.

Chapter 7

In this chapter the derivation of SPICE is revisited to streamline it and to provide further insights into this method and a new method named LIKES (likelihood-based estimation of sparse parameters) is obtained in a hyperparameter free manner from the maximum-likelihood principle applied to the same estimation problem as considered by SPICE. Through numerical simulations, it shown that both SPICE and LIKES provide accurate parameter estimates even from scarce data samples, with LIKES being more accurate than SPICE at the cost of an increased computational burden. The material in this chapter is based on:

- P. Stoica and P. Babu, SPICE and LIKES : Two hyperparameter-free methods for sparse-parameter estimation, Signal Processing, vol. 92, 1580-1590, 2012.

Chapter 8

In this chapter the grid selection problem for sparse estimation of spectral-line parameters is addressed. First a simple practical rule for choosing an initial value of gird size in a given situation is presented. Then the ways in which the estimation results corresponding to different values of grid size can be compared with one another and therefore how to select the “best” value of grid size

among those considered are addressed. Furthermore, a method for detecting when a grid is “too rough” and for obtaining refined parameter estimates in such a case is discussed. The material in this chapter is based on:

- P. Stoica and P. Babu, Sparse estimation of spectral lines: Grid selection problems and their solutions, *IEEE Transactions on Signal Processing*, vol. 60, pp. 962-967, 2012.

Chapter 9

In this chapter the problem of spectral-line analysis of nonuniformly sampled multivariate time series is addressed. SPICE and LIKES are re-derived for the considered model and their performances are compared numerically with that of a sparse method named multivariate sparse Bayesian learning (MSBL). The material in this chapter is based on:

- P. Babu and P. Stoica, Sparse spectral-line estimation for non uniformly sampled multivariate time series: SPICE, LIKES and MSBL, *20th European Signal Processing Conference*, Bucharest, Romania, 2012.

Chapter 10

In this chapter an estimation technique for analyzing radial velocity data commonly encountered in extrasolar planet detection is presented. The chapter starts with a discussion on the Keplerian model for radial velocity data measurements and it introduces a technique named the iterative adaptive approach (IAA) to estimate the 3D spectrum (power vs. eccentricity, orbital period and periastron passage time) of the radial velocity data. Then a discussion on different ways to regularize the estimation method in the presence of noise and measurement errors is included. A brief discussion on the computational aspects of the method is included. Finally, the significance of the spectral peaks is established by using a relaxation maximum likelihood algorithm (RELAX) and a generalized likelihood ratio test (GLRT). Numerical results presented at the end of this chapter include experiments carried out on both simulated and the real life data sets of the stars HD 63454, HD 208487 and GJ 876. The material in this chapter is based on:

- P. Babu, P. Stoica, J. Li, Z. Chen, and J. Ge, Analysis of radial velocity data by a novel adaptive approach, *The Astronomical Journal*, vol. 139, pp. 783-793, February 2010.
- P. Babu, P. Stoica, and J. Li, Modeling radial velocity signals for exoplanet search applications, *7th International Conference on Informatics in Control, Automation and Robotics*, Madeira, Portugal, 2010.

Chapter 11

In this chapter a discussion on the application of the SPICE algorithm to estimate the parameters of the Keplerian model commonly used in radial velocity data analysis for extrasolar planet detection is presented. The parameter estimates obtained from SPICE are then refined by means of a relaxation-based maximum likelihood algorithm (RELAX) and the significance of the resultant estimates is determined by a generalized likelihood ratio test (GLRT). Numerical experiments based on a real-life radial velocity data set of the star HD 9446 are included at the end of the chapter. The material in this chapter is based on:

- P. Babu and P. Stoica, A combined linear programming-maximum likelihood approach to radial velocity data analysis for extrasolar planet detection, 36th International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, 2011.

Chapter 12

In this chapter Sudoku puzzles are formulated and solved as a sparse linear system of equations. The chapter begins by showing that the Sudoku ruleset can be expressed as an underdetermined linear system: $Ax = b$, where A is of size $m \times n$ and $n > m$. It is then proved that the Sudoku solution is the sparsest solution of $Ax = b$, which can be obtained by ℓ_0 norm minimization problem. Instead of the ℓ_0 norm minimization problem, inspired by the sparse representation literature, a much simpler linear programming problem of minimizing the ℓ_1 norm of x is pursued and it is shown numerically that this approach solves representative Sudoku puzzles. The material in this chapter is based on:

- P. Babu, K. Pelckmans, P. Stoica, and J. Li, Linear Systems, Sparse Solutions, and Sudoku, IEEE Signal Processing Letters, vol. 17, no. 1, pp. 40-42, 2010.

List of papers not included in the thesis

Following is the list of papers that are not included in this thesis but were written during my PhD study. The papers included in the list below deal with problems like sinusoidal parameter estimation, model order selection, optimal experiment design and moving-average parameter estimation.

- P Babu and P Stoica, Comments on “Iterative estimation of sinusoidal signal parameters”. IEEE Signal Processing Letters, vol. 17, 1022-1023, 2010.

- P. Stoica and P. Babu, Algebraic derivation of Elfving theorem on optimal experiment design and some connections with sparse estimation, IEEE Signal Processing Letters, vol. 17, pp. 743-745, 2010.
- P. Stoica and P. Babu, The Gaussian data assumption leads to the largest Cramer-Rao bound, IEEE Signal Processing Magazine, vol. 28, pp.132-133, 2011.
- P. Stoica and P. Babu, On the proper forms of BIC for model order selection, IEEE Transactions on Signal Processing, In Press, 2012.
- P. Stoica and P. Babu, On the exponentially embedded family (EEF) rule for model order selection, IEEE Signal Processing Letters, In Press, 2012.
- P. Stoica and P. Babu, Model order estimation via penalizing adaptively the likelihood (PAL), IEEE Transactions on Signal Processing, Submitted, 2012.
- P. Babu, E. Gudmundson, and P. Stoica, Automatic cepstrum-based smoothing of the periodogram via cross-validation, 16th European Signal Processing Conference, Lausanne, Switzerland, 2008.
- P. Babu, E. Gudmundson, and P. Stoica, Optimal preconditioning for interpolation of missing data in a band-limited sequence, 42nd ASILO-MAR Conference on Signals, Systems and Computers, Pacific Grove, CA, 2008.
- P. Babu, P. Stoica, and T. Marzetta, An IQML type algorithm for AR parameter estimation from noisy covariance sequences, 17th European Signal Processing Conference, Glasgow, UK, 2009.
- B. Wahlberg, P. Stoica, and P. Babu, On estimation of cascade systems with common dynamics, 15th IFAC Symposium on System Identification, Saint-Malo, France, 2009.
- N. Sandgren, P. Stoica and P. Babu, On moving average parameter estimation, 20th European Signal Processing Conference, Bucharest, Romania, 2012.

1.2 Topics for future research

Following is a list of problems related to the topics discussed in the thesis that can be pursued in the future.

- Deriving necessary and sufficient conditions based on the average sampling rate for recovering a continuous time signal (or, its spectrum) from the sampled data is definitely an interesting research problem. Presently known conditions like Landau's condition on average sampling density are only necessary, so deriving necessary and sufficient conditions would be a worth topic to investigate.
- Most of the topics discussed in this thesis deal with the problem of analyzing the spectra of nonuniformly sampled data. It would also be interesting to look at the following problem: Given some prior knowledge about the spectrum (or the data) how to design an optimal sampling pattern for the stable recovery the signal from the sampled data? This has a lot of practical applications for example, in astronomical applications the cost of telescope usage could be expensive, where an optimal design of sample observation instances could reduce the experiment cost.
- Similar to the optimal sampling pattern design problem, designing optimal arrays for direction arrival estimation could also be pursued.
- Iterative adaptive algorithm (IAA) discussed in chapters 4, 5 and 10 was introduced in [122] and its local convergence properties are analyzed in [85]. However, little is known about its global convergence properties. It would be interesting to analyze the global convergence properties of IAA.
- SPICE and LIKES are among the most important contributions of this thesis. Both of them are empirically observed to be accurate, with LIKES being slightly more accurate than SPICE, but little is known about the statistical properties of the estimates of SPICE and LIKES. It would be worth pursuing the problem of deriving the expressions for mean square error of the SPICE and LIKES estimates.

Part I: Dense methods

2. Methods for signal reconstruction and spectral analysis from nonuniformly sampled data - a review

2.1 Introduction

The problem of spectral analysis of nonuniformly sampled data is well motivated both by its theoretical significance and by its wide spread application in fields like astronomy [86], seismology [7], paleoclimatology [87], genetics [57] and laser Doppler velocimetry [111]. The simplest way to deal with this problem is to compute the normal periodogram by neglecting the fact that data samples are nonuniform; this results in a dirty spectrum. Then a method called CLEAN [82] was proposed in the astronomical literature to do iterative deconvolution in the frequency domain to obtain the clean spectrum from the dirty one. A periodogram related method, which was also proposed in the astronomical literature is least squares periodogram (also called the Lomb-Scargle periodogram) [59] [86] which estimates the different sinusoids in the data by fitting them to the nonuniform data. Most recently, [122] introduced a new efficient method called Iterative Adaptive Approach (IAA), which relies on solving an iterative weighted least squares problem. In [106], consistent spectral estimation methods have been proposed by assuming the sampling patterns are known to be either uniformly distributed or Poisson distributed. Apart from these nonparametric methods there are also methods based on parametric modeling of the continuous time spectrum such as:

- maximum likelihood (ML) fitting of continuous time ARMA (CARMA) model to the nonuniformly sampled data [50],
- methods based on estimating the parameters of the continuous time model by approximating the derivative operators in the continuous time model [53],
- methods based on transforming the underlying continuous time model to an auxiliary discrete time model [65] and estimating the parameters of the continuous time model by identifying the parameters of the discrete time model.

There are also methods using the spectral analysis tools from uniform sampling domain by:

- obtaining the samples on an uniform grid by resampling and interpolation techniques,

- estimating the auto-covariance sequence of the data on a regular grid through slotting techniques [11] [42] or using suitable kernels [13] [41] [100],
- converting the nonuniform data problem into a missing data problem by techniques like slotted resampling, and estimating the underlying spectrum by either parametric modeling [49] [18] [19], or by nonparametric methods ([114], [97]).

Although the primary interest of this chapter is in the methods for spectral analysis of nonuniform data, a brief overview of the methods for exact reconstruction of the continuous time signal from the nonuniform data is also presented; once the signal is reconstructed, spectral analysis can be done on the reconstructed signal. These reconstruction methods rely on some assumptions on the underlying continuous time signal like the signal is assumed to be periodic bandlimited [33], bandlimited and sparse [15], known to be from a specific functional space like shift invariant space [4], etc. For example, using the assumption that the underlying signal is a periodic bandlimited signal with a known bandwidth, the continuous time domain signal can be effectively reconstructed from its finite nonuniform samples by solving a linear system. Together with the bandlimited assumption, if the signal is assumed to be sparse in the Fourier domain, then the continuous time signal can be reconstructed using tools from the recently developed field of compressed sensing or compressive sampling. The idea of signal reconstruction from nonuniform samples for signals from a specific functional space has its deep root in the area of functional analysis. For example, [32] proposes various iterative algorithms for the reconstruction of the continuous time signal from a finite number of nonuniform samples. Apart from this there are few signal reconstruction methods [112] [70] specific to sampling patterns like periodic nonuniform sampling, which is common in high speed analog to digital converters (ADC). In [112], the authors have given formulas for exact reconstruction of the signals sampled in a periodic nonuniform fashion, under the assumption that the underlying signals are multiband with a known band structure. The paper [70] has dealt with the same problem but without the knowledge of the band structure.

Although the literature for nonuniformly sampled data is quite rich and diverse, the tools and techniques employed by different methods are not classified and compared. One of the main goals of this chapter is to discuss and classify the different methods based on:

- the signal model (nonparametric or parametric),
- the sampling pattern,
- the spectrum of the signal (discrete or continuous).

The different methods are then compared based on their performance on simulated or real life nonuniform data sets.

The outline of the chapter is as follows. In section 2.2, we will start with the various notations used in the paper, and discuss preliminaries like various signal models, and sampling patterns, and include tools like least squares,

maximum likelihood, resampling and interpolation techniques, which will be used in the later sections of this chapter.

In section 2.3, we present various methods for exact signal reconstruction from nonuniform samples and compare their complexities. We end that section by including a subsection about the choice of parameters like Nyquist limit, frequency resolution and resampling time in the nonuniform data case.

Section 2.4 deals with the different methods for spectral analysis of nonuniformly sampled data, which are classified based on the three factors: signal model, sampling pattern and the underlying spectrum. We compare the different methods discussed in this section by carrying out numerical simulations on two real life data sets.

Section 2.5 contains the conclusions of the chapter.

2.2 Notations and Preliminaries

The following table contains some notations often used in this chapter.

Notations	
$f(t)$	continuous time signal
t_s	sampling period
$f(nt_s)$	uniformly sampled signal
$f^*(\cdot)$	complex conjugate of $f(\cdot)$
$F(\cdot)$	Fourier transform of $f(t)$
$r(\tau)$	autocovariance sequence
$\Phi(\cdot)$	power spectrum
Ω, ω	analog/digital frequency
$\{t_k\}_{k=1}^N$	nonuniform sampling time instants
$E(\cdot)$	expectation operation
$p_X(x)$	PDF of a random variable X
$\ \cdot\ _2$	l_2 norm
$\ \cdot\ _1$	l_1 norm
$(\cdot)^H$	conjugate transpose
W^\dagger	$(W^H W)^{-1} W^H$
$\xrightarrow{\mathcal{F}}$	Fourier transfrom
$\xrightarrow{\mathcal{F}^{-1}}$	Inverse Fourier transfrom.

2.2.1 Signal models

In this subsection, we will introduce the signal models, both continuous and discrete time, used in this chapter.

Discrete time models:

The discrete time stationary stochastic signal, $f(nt_s)$, can be modeled by a general equation of the form:

$$\sum_{p=0}^P a_p f(n-p) = \sum_{m=0}^M b_m e(n-m) + \sum_{l=0}^L c_l u(n-l) \quad \forall n \quad (2.1)$$

where $e(n)$ represents a zero mean, unit variance white noise, and $u(n)$ represents another stationary process which is independent of $e(n)$, the t_s in $f(nt_s)$ has been omitted for notational simplicity.

- If the coefficients, $\{c_l\}_{l=0}^L$, $\{b_m\}_{m=1}^M$ are set to zero, then equation (2.1) represents a autoregressive process (AR) of order P .
- If the coefficients, $\{c_l\}_{l=0}^L$, $\{a_p\}_{p=1}^P$ are set to zero, and $a_0 = 1$, then equation (2.1) represents an moving average process (MA) of order M .
- If the coefficients, $\{b_m\}_{m=1}^M$ are set to zero, then equation (2.1) represents an autoregressive process with exogenous input (ARX) of order (P, L) .

- If the coefficients, $\{c_l\}_{l=0}^L$ are set to zero, then equation (2.1) represents an ARMA(P, M).

Continuous time models:

The ARMA model can be generalized to the continuous time case as shown below:

$$(a_0 p^M + a_1 p^{M-1} + \dots + a_M) f(t) = (b_0 p^N + b_1 p^{N-1} + \dots + b_N) e(t). \quad (2.2)$$

where p denotes the differentiation operator, $\frac{d}{dt}$, and $e(t)$ denotes the continuous time white noise of unit intensity. Similarly to the discrete time case, if the coefficients, $\{b_p\}_{p=0}^{N-1}$ are set to zero, then equation (2.2) represents a continuous time autoregressive process (CAR(M)), else it represents a continuous time ARMA (CARMA(M, N)) process.

2.2.2 Sampling patterns

In this subsection, we will describe various sampling patterns that are used in the other sections.

Uniform sampling

The sampling instants are uniformly placed, i.e, for any k we have that $t_k = kt_s$, where t_s is the sampling period and it is given by $\frac{\pi}{\Omega_{max}}$, where Ω_{max} is the maximum frequency present in the continuous time signal.

Sampling pattern with specific distribution

In some cases the sampling instants are random and follow some specific probability distribution like Poisson, uniform distribution, or stratified uniform distribution.

- *Uniform distribution*

The PDF of uniformly distributed sampling instants within the interval of $[0, T]$ is given by:

$$p(t) \triangleq \begin{cases} \frac{1}{T} & \text{when } t \in [0, T] \\ 0 & \text{when } t \notin [0, T]. \end{cases} \quad (2.3)$$

- *Stratified uniform distribution*

As in the uniform distribution case, here too the sampling instants are uniformly distributed, but stratified. Let $0 = \tau_0 < \tau_1 < \dots < \tau_N = T$, denote an N partition of the interval $[0, T]$ defined by a probability density function, $h(t)$, given by:

$$\int_0^{\tau_j} h(t) dt = \frac{j}{N}, \quad j = 0, 1, \dots, N. \quad (2.4)$$

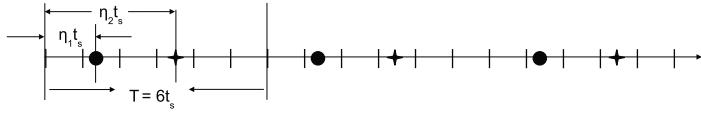


Figure 2.1. Periodic nonuniform sampling.

Then the sampling points $\{t_k\}_{k=1}^N$ are selected such that each t_k is uniformly distributed in the subinterval $[\tau_{k-1}, \tau_k]$.

- *Poisson distribution*

Here any sampling instant t_k , will be modeled in terms of the previous sampling instant as shown below:

$$\begin{aligned} t_0 &= 0 \\ t_k &= t_{k-1} + \gamma_k \quad k = 1, 2, \dots, \end{aligned} \tag{2.5}$$

where $\gamma_k \quad k = 1, 2, \dots$, are *i.i.d.* random variables with a common exponential distribution $p(\tau) = \beta \exp(-\beta \tau)$, where β defines the mean sampling rate. This leads to the spacing between the sampling instants, given by $(t_{s+k} - t_s)$, following Poisson distribution with PDF given by:

$$p_k(x) = \begin{cases} \beta \frac{(\beta x)^{k-1}}{(k-1)!} \exp(-\beta x) & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{2.6}$$

Periodic nonuniform sampling

Here the regular sampling grid with sampling time t_s is divided into periods of size $T = Mt_s$, where $M \in \mathbb{Z}_+$. Within each period, the sampling points are nonuniformly placed and they can be written as

$$\begin{aligned} t_k &= nT + \eta_k t_s & k = 1, \dots, K \\ \text{with } & 0 \leq \eta_k < M, n \in \mathbb{Z} \end{aligned} \tag{2.7}$$

Figure 2.1 shows a periodic nonuniform sampling grid for $M = 6, K = 2$.

Arbitrary sampling

We also consider arbitrary sampling schemes, which cannot be characterized by any simple probability distribution.

2.2.3 Resampling and interpolation techniques

We will now discuss resampling and interpolation techniques used in the later sections. They are generally employed to interpolate the data on a uniform grid from nonuniformly placed samples.

Kernel based interpolation

Given the N nonuniformly placed samples, $\{y(t_k)\}_{k=1}^N$, of the continuous time signal $y(t)$, within the interval $[0, T]$ the signal $y(t)$ can be interpolated by

$$y_i(t) = \sum_{k=1}^N y(t_k) K(t, t_k) \quad t \in [0, T] \quad (2.8)$$

where $y_i(t)$ is the interpolated signal and $K(\cdot)$ denotes the interpolation kernel. Some of the commonly used kernels are

- *Sinc kernel*

$$K(t, t_k) = \frac{\sin(\frac{\pi(t-t_k)}{b_1})}{\pi(t-t_k)} \quad (2.9)$$

where b_1 determines the mainlobe width of the sinc.

- *Gaussian kernel*

$$K(t, t_k) = \frac{1}{\sqrt{2\pi}b_2} \exp\left(\frac{-(t-t_k)^2}{2b_2^2}\right) \quad (2.10)$$

where b_2 determines the bandwidth of the Gaussian function.

- *Laplacian kernel*

$$K(t, t_k) = \frac{1}{2b_3} \exp\left(\frac{-|t-t_k|}{b_3}\right) \quad (2.11)$$

where b_3 determines the bandwidth of the Laplacian function.

- *Rectangular kernel*

$$K(t, t_k) = \begin{cases} \frac{1}{2b_4} & |t-t_k| \leq b_4 \\ 0 & \text{otherwise.} \end{cases} \quad (2.12)$$

where b_4 determines the width of the rectangular window. In the literature, interpolation through a rectangular kernel is commonly called slotting technique.

For all these kernels, the user must make the right choice of the parameters. Furthermore none of the above mentioned kernel based interpolation techniques has the interpolation property: $y_i(t_k) = y(t_k)$ $k = 1, 2, \dots, N$.

Nearest neighbor interpolation

In this case, the time interval $[0, T]$ is divided into a regular grid with spacing between the sampling instants equal to t_r , the choice of t_r is discussed in section 2.3. Then the sample value at any point mt_r , is assigned the data value which is closer to mt_r .

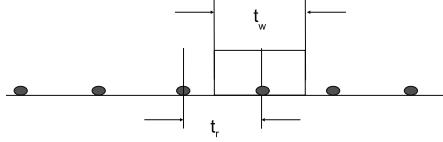


Figure 2.2. Grids for slotted resampling technique.

Slotted resampling

As for the nearest neighbor interpolation, the time interval $[0, T]$ is divided into a regular grid with spacing t_r , and around each point, a slot of width t_w is placed (a reasonable choice for t_w is equal to $\frac{t_r}{2}$). Then the sample value at any point mt_r is assigned the nearest sampled value within the slot around mt_r . If there is no sampled value falling within a slot then the corresponding regular grid point is not assigned a sample value, which leads to a missing data problem.

2.2.4 Least Squares, Maximum Likelihood

In this subsection, we will briefly describe some estimation techniques that we will use in the remaining sections. Let $\{y_i\}_{i=1}^N$ denote the N measurements of an experiment with a model

$$y_i = a_i^H x + n_i \quad i = 1, \dots, N \quad (2.13)$$

where x , of size $M < N$, denotes the parameters of interest and n_i denotes the noise in the measurements.

Least Squares (LS)

The least-squares (LS) estimate of x is given by

$$\hat{x}_{LS} = \arg \min_x \|y - Ax\|_2^2. \quad (2.14)$$

where $y = [y_1, \dots, y_N]^T$, and $A = [a_1, \dots, a_N]^H$ is assumed to have full column rank. Then the solution to the minimization problem is analytically given by $\hat{x}_{LS} = A^\dagger y$. If the covariances of the noise in the measurements are known, then an optimal solution can be obtained by solving a weighted least squares (WLS) problem:

$$\begin{aligned} \hat{x}_{WLS} &= \arg \min_x \|y - Ax\|_{R^{-1}}^2 \\ &= \arg \min_x (y - Ax)^H R^{-1} (y - Ax) \\ &= (A^H R^{-1} A)^{-1} A^H R^{-1} y. \end{aligned} \quad (2.15)$$

where R represents the covariance matrix of the noise.

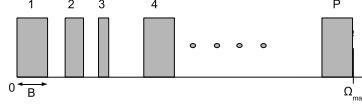


Figure 2.3. Multiband spectrum.

Maximum Likelihood (ML)

The maximum likelihood (ML) estimate of x is obtained by maximizing the likelihood function $L(x)$:

$$\begin{aligned}\hat{x}_{ML} &= \arg \max_x \{L(x) \stackrel{\Delta}{=} p(y/x)\} \\ &= \arg \max_x \frac{1}{(\pi)^N |R|} e^{-(y-Ax)^H R^{-1} (y-Ax)}.\end{aligned}\quad (2.16)$$

where $p(y/x)$ represents the PDF of the measurements. The second equality in the above equation follows from the assumption that the observations are jointly Gaussian distributed. So in this case, ML and WLS give the same estimate.

2.3 Signal Reconstruction Methods

In this section, we describe various methods for signal reconstruction from nonuniform samples. The table 2.1 summarizes some of the known exact signal reconstruction methods. In the table, Ω_{max} and the set $S = \text{supp}(F(\Omega))$ denote the maximum frequency in the spectrum and the support of the spectrum respectively, $|S|$ denotes the cardinality of the set S , and $L^2(\cdot)$ denotes the vector space of square integrable functions. The frame $\{\phi_k\}$ of a vector space V denotes a set of linearly dependent vectors spanning V , which are such that $\forall v \in V$ there exist c and C with $0 < c \leq C < \infty$ such that

$$c\|v\|^2 \leq \sum_k |v^H \phi_k|^2 \leq C\|v\|^2 \quad (2.17)$$

where $v^H \phi_k$ denotes the inner product; furthermore for any $v \in V$ the dual frames $\{\tilde{\phi}_k\}$ satisfy

$$\sum_k (v^H \phi_k) \tilde{\phi}_k = \sum_k (v^H \tilde{\phi}_k) \phi_k = v. \quad (2.18)$$

As can be seen from the table the methods either require an infinite number of samples or some restrictive assumptions on the signal to exactly recover the signal. However, in most practical situations, we have only a finite number of nonuniform samples of the data corrupted with noise, and little or no prior

Exact signal reconstruction methods					
Method	Assumptions	Descriptions	Computational complexity	Comments	
Lagrange like interpolation [44] [67]	$ t_k - k\delta_s < \frac{\delta_s}{4} \quad \forall k \in \mathbb{Z}$ and $\delta_s = \frac{\pi}{\Omega_{max}}$.	$f(t) = \sum_{k=-\infty}^{\infty} f(t_k) \frac{L(t)}{L(t_k)(t-t_k)}$ $L(t) = (t - t_0) \prod_{k=1}^{\infty} \left(1 - \frac{t}{t_k}\right) \left(1 - \frac{t}{t_{-k}}\right)$	Requires infinite number of flops.	Requires infinite number of samples for exact signal reconstruction.	
Frame based reconstruction [44] [67]	$\{e^{it_k \Omega}\}$ is a frame for $L^2(-\Omega_{max}, \Omega_{max})$ $S \subseteq [-\Omega_{max}, \Omega_{max}]$.	$f(t) = \sum_{k=-\infty}^{\infty} f(t_k) R_k(t)$ $R_k(t) = \int_{-\Omega_{max}}^{\Omega_{max}} \psi_k(\Omega) e^{it \Omega} d\Omega$, where $\{\psi_k(\Omega)\}$ is the dual frame.	Requires infinite number of flops.	Requires infinite number of samples for exact signal reconstruction.	
Sparse approach [15]	$S \subseteq [-\Omega_{max}, \Omega_{max}]$ and $f(t)$ is p sparse i.e. $ S \leq p$, and $N > p$.	$\min_f \ f\ _1 \quad \text{s.t. } f = \Phi f$ where $f = [f(t_1), \dots, f(t_N)]^T$, $[F(\Omega_K); \dots, F(\Omega_K)]^T$, $\Omega_k = (e^{it_n \Omega_k}), K > N$.	Requires solving a Linear Program (LP), which can be solved efficiently by well developed softwares [39] [104] in polynomial time.	The sparsity p should be known. It requires only finite time samples.	
Linear system approach [33]	$f(t)$ is periodic with period T and it is also bandlimited with a bandwidth of K , and $N \geq 2K + 1$.	$f(t) = \sum_{k=K}^{-K} F(k) e^{ikt \frac{2\pi}{T}} \quad t = t_1, \dots, t_N$ $f = W \tilde{f} \Leftrightarrow \tilde{f} = W^\dagger f$	Requires solving a linear system and it requires $O(N^3)$ flops.	The bandwidth K should be known apriori.	
Multiband spectrum reconstruction [112][70]	$f(t)$ is a multiband signal with no more than P bands of maximum size B as shown in figure 2.3.	$F_{\eta_k}(\Omega) = \frac{1}{M_k} \sum_{r=0}^{M_k-1} e^{i \frac{2\pi}{M_k} \eta_k r} F\left(\Omega + \frac{2\pi r}{M_k}\right)$ where $\Omega \in \Omega_0 = \left[0, \frac{2\pi}{M_k}\right)$, $1 \leq k \leq K$. $F(\Omega)$ denotes the Fourier transform of $f(t)$.	If the band structure is known, then for each $\Omega \in \Omega_0$, the method requires $O(K^3)$ flops to solve the linear system.	The linear system can be solved only when $f(\Omega)$ is K sparse for all $\Omega \in \Omega_0$.	
	$\{\eta_k\}$ is a periodic nonuniform sampling pattern and the sampling instants in each period are given by $t_k = nM_{\delta_s} + \eta_k t_{\delta_s} \quad k = 1, \dots, K$ and $0 \leq \eta_k < M$	$\Leftrightarrow \begin{cases} \tilde{f}(\Omega) = A f(\Omega) & \forall \Omega \in \Omega_0 \\ \tilde{f}(\Omega) = [F_{\eta_1}(\Omega), \dots, F_{\eta_K}(\Omega)]^T, & \\ f_j(\Omega) = F\left(\Omega + \frac{2\pi j}{M_k}\right) \text{ and } A_{jk} = \frac{1}{M_k} \left(e^{i \frac{2\pi}{M_k} \eta_j \omega}\right) \end{cases}$	If the band structure is unknown, then for each $\Omega \in \Omega_0$, the method requires to solve a LP as in sparse approach.		

Table 2.1. Exact signal reconstruction methods.

knowledge about the signal spectrum, and this makes these methods somewhat impractical. In the next section, we will describe various methods that can handle practical data.

Before we end this section, we will discuss the choice of various parameters like Nyquist frequency, frequency resolution and resampling time that are needed for spectral estimation and interpolation of non-uniformly sampled data. For a given finite number of nonuniform samples, Nyquist frequency indicates the rollover frequency beyond which the spectrum replicates; frequency resolution determines the minimum frequency spacing that can be resolved; and the resampling time determines the uniform grid spacing over which the nonuniform data can be resampled or interpolated.

2.3.1 Nyquist frequency (Ω_{nus})

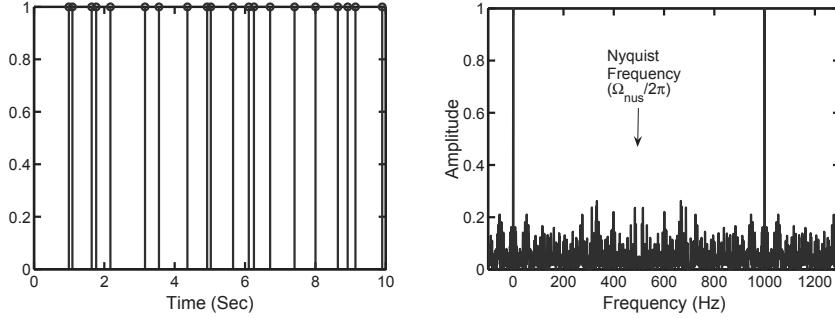
Nyquist frequency for uniformly sampled signals is well defined through the Nyquist-Shannon sampling theorem. However, for non-uniformly sampled signals, there is no well accepted definition. For example, [86] [82] define the Nyquist frequency in the nonuniform case as $1/2t_m$, where t_m is the smallest time interval in the data, while [100] defines it as $1/2t_a$, where t_a is the average sampling rate. However the maximum frequency that can be inferred from nonuniform samples appears to be larger. In [31] (see also [100]), a more realistic way of defining the Nyquist or rollover frequency for nonuniform samples has been described. Given N nonuniform samples with sampling instants $\{t_k\}_{k=1}^N$, the spectral window at any frequency Ω is defined as

$$W(\Omega) = \left| \frac{1}{N} \sum_{k=1}^N e^{i\Omega t_k} \right|^2. \quad (2.19)$$

We can easily verify that $W(0) = 1$, and for any $\Omega \neq 0$, $W(\Omega) \leq 1$. In the case of uniform sampling, the spectral window will be a periodic function with period equal to twice the Nyquist frequency, that is for any integer m and Ω , $W(\Omega) = W(\Omega + 2m\Omega_{us})$, where Ω_{us} is the Nyquist frequency for the uniform sampling case. Similarly the Nyquist frequency (Ω_{nus}) for nonuniform data can be defined as the smallest frequency for which $W(2\Omega_{nus}) \approx 1$.

2.3.2 Frequency resolution ($\Delta\Omega$)

Frequency resolution determines the minimum frequency spacing between two sinusoids which can be resolved. In the uniform sampling case, for a given set of N uniform samples, the frequency resolution, $\Delta\Omega$, is given by $\frac{2\pi}{(t_N - t_1)} = \frac{2\pi}{Nt_s} = \frac{4\pi}{Nt_{us}} = \frac{2\Omega_{us}}{N}$, where t_s and t_{us} represents uniform sampling and Nyquist time intervals respectively. Similarly for the nonuniform sampling case, the frequency resolution can be defined as $\frac{2\pi}{t_N - t_1}$.



(a) Nonuniform Samples

(b) Spectral Window

Figure 2.4. The sampling pattern and the spectral window.

Spectral parameter	Value
Average sampling time (t_a) and $\Omega_a/2\pi$	0.4688 sec, 1.06Hz
Minimum time interval (t_m) and $\Omega_m/2\pi$	0.0990 sec, 5.05 Hz
Resampling time (t_r) and $\Omega_{nus}/2\pi$	0.001 sec, 500 Hz

Table 2.2. Spectral parameters.

2.3.3 Resampling time (t_r)

The resampling time, denoted by t_r , determines the minimum spacing of the uniform grid over which the nonuniformly sampled data can be resampled or interpolated. It is given by $\frac{\pi}{\Omega_{nus}}$.

As an example, we have generated 20 nonuniform sampling times in the interval $[0 - 10]$ s. The sampling pattern and its spectral window are shown in figure 2.4. Table 2.2 shows the values of the minimum time interval, average time interval and resampling time. It can be seen from the table that the Nyquist frequency defined from the t_m or from t_a is much smaller than the Nyquist frequency obtained from the spectral window. The symbols Ω_m and Ω_a in the table 2.2 denote the Nyquist frequencies calculated according to t_m and t_a respectively.

2.4 Spectral Analysis Methods

In the first part of this section, we will describe different methods available in the literature for spectral analysis of nonuniform data and classify them based on the following:

- the signal model (nonparametric or parametric),
- the sampling pattern,
- the spectrum of the signal (discrete or continuous).

The performance of different methods on both simulated and real life nonuniform data sets is analyzed in the second part of this section. The methods will be described under four broad categories: methods based on least squares; methods based on interpolation techniques; methods based on slotted resampling; methods based on continuous time models.

2.4.1 Methods based on least squares

In this subsection, we will describe spectral analysis methods for nonuniform data which are based on least squares. Let $\{f(t_k)\}_{k=1}^N$ denote the sequence of nonuniformly spaced observations of $f(t)$. The frequency interval $[0, \Omega_{nus}]$ is divided into M equally spaced points with spacing between them equal to $\Delta\Omega$, where $M = \lfloor \frac{\Omega_{nus}}{\Delta\Omega} \rfloor$, and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x . Any frequency on the grid can be denoted by $\Omega_m = m\Delta\Omega$ for some integer $m \in [0, M]$.

Schuster Periodogram

The classical or Schuster periodogram at any frequency Ω_m , denoted here by $S_p(\Omega_m)$, is a solution to the following least squares problem:

$$\begin{aligned}\hat{\beta}_m &= \arg \min_{\beta_m} \sum_{k=1}^N |f(t_k) - \beta_m e^{i\Omega_m t_k}|^2; \\ S_p(\Omega_m) &= |\hat{\beta}_m|^2 = \frac{1}{N^2} \left| \sum_{k=1}^N f(t_k) e^{-i\Omega_m t_k} \right|^2.\end{aligned}\quad (2.20)$$

Least-squares (also called Lomb-Scargle) Periodogram

The LS periodogram at any frequency Ω_m , can also be expressed as a least squares problem [59]:

$$\min_{\substack{\alpha > 0 \\ \phi \in [0, 2\pi]}} \sum_{k=1}^N [f(t_k) - \alpha \cos(\Omega_m t_k + \phi)]^2. \quad (2.21)$$

The data here is assumed to be real valued as the LS periodogram has been defined only for the real valued data. Using $a = \alpha \cos(\phi)$ and $b = -\alpha \sin(\phi)$ in (2.21), we get

$$\min_{a,b} \sum_{k=1}^N [f(t_k) - a \cos(\Omega_m t_k) - b \sin(\Omega_m t_k)]^2. \quad (2.22)$$

Introducing the following notations

$$\begin{aligned} R &= \sum_{k=1}^N \begin{bmatrix} \cos(\Omega_m t_k) \\ \sin(\Omega_m t_k) \end{bmatrix} \begin{bmatrix} \cos(\Omega_m t_k) & \sin(\Omega_m t_k) \end{bmatrix}, \\ r &= \sum_{k=1}^N \begin{bmatrix} \cos(\Omega_m t_k) \\ \sin(\Omega_m t_k) \end{bmatrix} f(t_k), \end{aligned} \quad (2.23)$$

the solution to the minimization problem in (2.22) can be written as

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = R^{-1}r. \quad (2.24)$$

The power spectral estimate, denoted by $S_{LS}(\cdot)$, is given by:

$$\begin{aligned} S_{LS}(\Omega_m) &= \frac{1}{N} \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix} R \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} \\ &= \frac{1}{N} r^T R^{-1} r. \end{aligned} \quad (2.25)$$

Iterative adaptive approach (IAA)

Following [122], let us introduce the following notations

$$f = \begin{bmatrix} f(t_1) \\ \vdots \\ f(t_N) \end{bmatrix}; a_m = \begin{bmatrix} e^{i\Omega_m t_1} \\ \vdots \\ e^{i\Omega_m t_N} \end{bmatrix} \quad (2.26)$$

Using the above notations, a WLS fitting criterion, whose use is the central idea of the IAA algorithm, can be defined as follows:

$$\min_{\beta_m} \|f - a_m \beta_m\|_{R^{-1}}^2 \quad (2.27)$$

where β_m represents the amplitude of the frequency Ω_m and R represents an estimate of the covariance matrix, given by $R = \sum_{m=1}^M |\beta_m|^2 a_m a_m^H$. The solution to the above problem is given by:

$$\hat{\beta}_m = \frac{a_m^H R^{-1} f}{a_m^H R^{-1} a_m}. \quad (2.28)$$

Since the covariance matrix R depends on β_m , an iterative algorithm is used to estimate β_m and R . The steps in the IAA algorithm can be tabulated as shown in the table 2.3.

Numerical simulations are carried out to compare the methods based on least squares analysis. The data used here is simulated by picking samples from a mix of three frequencies 10, 15.5 and 16 Hz, with amplitudes 2, 3

Initialization

Use the estimate in (2.28) with $R = I$ as the initial value.

Iteration

At the i^{th} iteration, the estimate of β at any frequency Ω_m is given by:

$$\hat{\beta}_m^i = \frac{a_m^H(R^i)^{-1}f}{a_m^H(R^i)^{-1}a_m} \quad \text{for } m = 1, \dots, M$$

$$\text{where } R^i = \sum_{m=1}^M |\hat{\beta}_m^{i-1}|^2 a_m a_m^H.$$

Termination

The iteration will be terminated when the relative change in $\hat{\beta}_m$, $|\hat{\beta}_m^i - \hat{\beta}_m^{i-1}|^2$, is less than 10^{-4} .

The final spectral estimate at frequency Ω_m , denoted by $S_{IAA}(\Omega_m)$, is given by:

$$S_{IAA}(\Omega_m) = |\hat{\beta}_m^c|^2,$$

where $\hat{\beta}_m^c$ denotes the estimate obtained at convergence.

Table 2.3. Iterative adaptive approach algorithm (IAA).

and 4 respectively in the presence of white noise. The sampling pattern is generated according to Poisson distribution as described in the section 2.2.2. Figure 2.5a shows the time instants of the 20 generated samples. It can be seen from the figures (2.5b), (2.5c) and (2.5d), that IAA works better than Schuster and LS periodogram: IAA clearly shows the peak at 10 Hz, while this peak is lost in the background noise in the periodogram based methods; IAA also clearly resolves the two closely spaced frequencies at 15.5 Hz and 16 Hz.

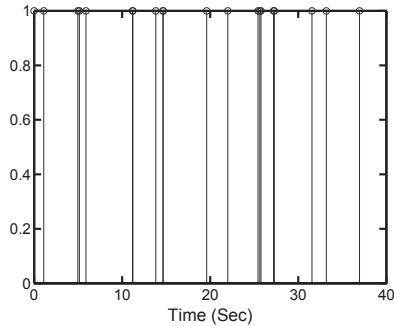
2.4.2 Methods based on interpolation

In this subsection, we describe different methods for spectral analysis that are based on interpolation in the lag domain. The primary interest of this chapter in analyzing the power spectra justifies the restriction to interpolation methods in the lag domain (however the methods discussed below are also applicable to the sample domain). Once the covariance values are obtained on a regular grid through interpolation, spectral analysis methods for the uniform sampling case can be applied to them.

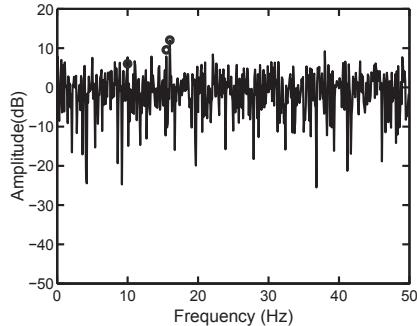
As described in the subsection 2.2.3, given a set of samples $\{f(t_k)\}_{k=1}^N$ of $f(t)$ within a interval $[0, T]$, various kernels can be used to interpolate the covariance sequence ($r(\tau)$) of $f(t)$. The general structure of kernel based covariance interpolators is given by:

$$r_I(\tau) = \sum_{l=1}^N \sum_{k=1}^N f(t_l) f^*(t_k) K(\tau, t_l - t_k) \quad \tau \in [0, T] \quad (2.29)$$

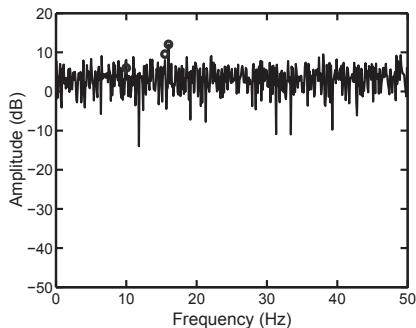
where $r_I(\tau)$ denotes the interpolated covariance sequence. For example, Gaussian, Laplacian [13], sinc [100], or rectangular kernels [11] [42] can be used to interpolate the covariance function. However in all these methods, the user has to make the right choice of the kernel parameters like bandwidth, mainlobe



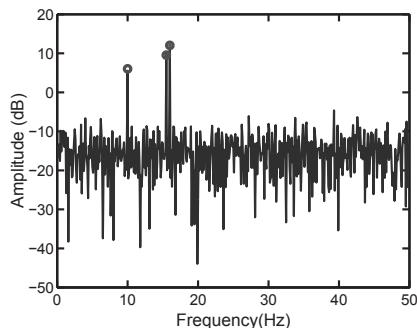
(a) Sampling pattern



(b) Schuster periodogram



(c) LS periodogram



(d) IAA spectrum

Figure 2.5. a) Sampling time instants of 20 samples. b) Schuster periodogram c) LS periodogram. d) IAA spectrum, and the circles denote the amplitudes of the true frequencies, 10, 15.5 and 16 Hz, present in the signal. The signal to noise ratio is 17 dB, and the frequency spacing is 0.1Hz.

width and window width. A further drawback of these kernel based covariance interpolators, except the one based on the sinc kernel, is that they do not ensure a positive semidefinite covariance sequence, i.e the power spectrum obtained from the interpolated covariance sequence could be negative at some frequencies. To ensure the positive semidefiniteness of the interpolated covariance sequence, each estimator has to carry out an additional step as shown below.

Let $\Phi_I(\Omega)$ denote the power spectral density obtained from $r_I(\tau)$; then a positive semidefinite covariance sequence, denoted by $r_P(\tau)$, can be obtained as:

$$\begin{aligned} r_I(\tau) &\xrightarrow{\mathcal{F}} \Phi_I(\Omega) \\ \Phi_P(\Omega) &= \begin{cases} \Phi_I(\Omega) & \Phi_I(\Omega) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \\ \Phi_P(\Omega) &\xrightarrow{\mathcal{F}^{-1}} r_P(\tau) \end{aligned} \quad (2.30)$$

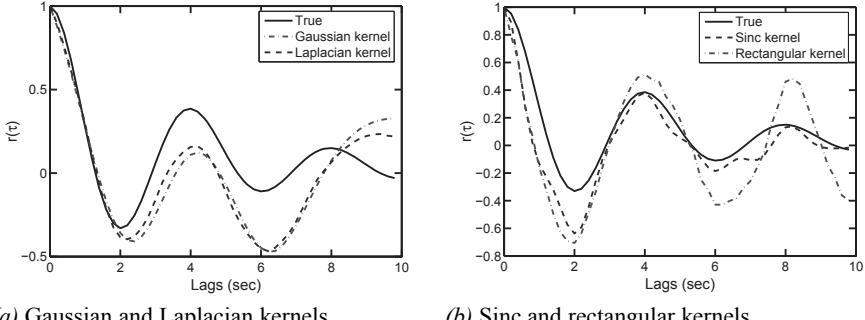
Numerical results are carried out to compare the various kernels used for covariance interpolation. A CARMA(3,2) has been simulated with AR and MA polynomials given by $a(s) = s^3 + 0.7s^2 + 2.629s + 0.506$ and $b(s) = s^2 + 5s + 2.792$ respectively. For numerical simulations, 50 nonuniformly spaced samples within a time interval of $[0 - 10]$ seconds are generated, see [100] for more details on CARMA process simulation. On those 50 samples of the process, four kernel based methods for covariance interpolation are applied. Figure 2.6 shows the interpolated sequence over the interval $[0 - 10]$ seconds; as can be seen from figure 2.6a the Gaussian and Laplacian kernels with bandwidths equal to 1 second give almost the same interpolated sequences. The plots in figure 2.6b show the interpolated covariance sequence obtained using the sinc and rectangular kernels with corresponding bandwidth and slot width equal to $4t_a$ and $2t_a$ respectively, where t_a represents the average sampling period. The bandwidth and slot width of the kernels are chosen arbitrarily for this simulation, as to our knowledge there is no optimal way of choosing them available in the literature.

Equispaced covariance lags can be obtained from $r_P(\tau)$ by sampling it over an uniform grid, and then any spectral analysis method designed for uniform sampling can be applied to the so-obtained uniform covariance sequence; the interested reader is referred [100] for details. Table 2.4 summarizes how the different uniform sampling methods (parametric or nonparametric) can be applied to the data obtained from different interpolation techniques.

2.4.3 Methods based on slotted resampling

ML fitting of AR model to nonuniform samples

Given a set of nonuniform samples within the interval $[0, T]$, the technique of slotted resampling, as described in the section 2.2.3, can be applied to obtain



(a) Gaussian and Laplacian kernels

(b) Sinc and rectangular kernels

Figure 2.6. Interpolated covariance sequence with a) Gaussian kernel of bandwidth $b_2 = 1$ sec and Laplacian kernel of bandwidth $b_3 = 1$ sec. b) Sinc kernel of main-lobe width $b_1 = 4t_a$ and rectangular kernel of slot-width $b_4 = 2t_a$.

Interpolation type	Data interpolation (SR, NN and KDI)	Covariance interpolation (KCI)
Uniform sampling method	<p>(Interpolated data sequence $\{f(nt_r)\}_{n=1}^{\tilde{N}}$ is obtained from $\{f(t_n)\}_{n=1}^N$. From $\{f(nt_r)\}$, an estimate of sample covariance matrix R_{DI} is obtained. In the case of SR, the missing sample values in $\{f(nt_r)\}$ are assumed to be zero).</p>	<p>(Interpolated covariance sequence $\{r(n\tau_r)\}_{n=1}^{\tilde{N}}$ is obtained from $\{f(t_n)\}_{n=1}^N$. A Toeplitz covariance matrix R_{CI} is then formed from $\{r(n\tau_r)\}_{n=1}^{\tilde{N}}$).</p>
Periodogram	$\frac{1}{N^2} \left \sum_{k=1}^{\tilde{N}} f(kt_r) e^{-j\Omega k t_r} \right ^2.$	$a^H R_{CI} a, a = \frac{1}{\sqrt{N}} [e^{j\Omega \tau_r}, \dots, e^{j\Omega \tilde{N} \tau_r}]^T.$
Capon	$\frac{1}{a^H R_{DI}^{-1} a}, a = \frac{\sqrt{2}}{\sqrt{N}} [e^{j\Omega \tau_r}, \dots, e^{j\Omega \tilde{N} \tau_r}]^T.$	$\frac{1}{a^H R_{CI}^{-1} a}, a = \frac{1}{\sqrt{N}} [e^{j\Omega \tau_r}, \dots, e^{j\Omega \tilde{N} \tau_r}]^T.$
Subspace methods (MUSIC, ESPRIT) [98]	<p>The subspace based spectral estimates are obtained from R_{DI}.</p>	<p>The subspace based spectral estimates are obtained from R_{CI}.</p>

Table 2.4. Various combinations of uniform sampling methods and interpolation techniques; t_r represents the resampling time and Ω lies in the interval $\left[0, \frac{\pi}{t_r}\right]$ (SR-Slotted Resampling, NN- Nearest Neighbor interpolation, KCI- Kernel based Covariance Interpolation, KDI- Kernel based Data Interpolation).

samples on an uniform grid with some missing samples. In [49] [18] and [19], various methods have been proposed for maximum likelihood fitting of a discrete time AR model to time series with missing observations. In [49], an approach based on state space modeling has been used to obtain the ML estimate of AR parameters, which will be referred here as ML-Jones, whereas in [19] an approximate but faster method named autoregressive finite interval likelihood (ARFIL) has been proposed.

Let $\{f(t_k)\}_{k=1}^N$ denote the sequence of nonuniformly spaced observations of a stationary process sampled within the time interval $[0, T]$, and let $\{\hat{f}(nt_r)\}_{n=1}^{\tilde{N}}$ represent the sequence obtained after slotted resampling over an uniform grid of resolution t_r , where in general $\tilde{N} \geq N$. The resampled sequence is assumed to be obtained from a discrete time zero mean, Gaussian distributed AR process of order K , such that

$$\hat{f}(n) + a_1\hat{f}(n-1) + \cdots + a_K\hat{f}(n-K) = \varepsilon(n). \quad (2.31)$$

where $\varepsilon(n)$ denotes an white Gaussian noise with zero mean and varaince σ_ε^2 , and $\{a_k\}_{k=1}^K$ are the AR parameters. In the above equation, t_r in the resampled sequence has been dropped for notational simplicity. The Yule-Walker equations for the given AR(K) are given by:

$$r(n) + a_1r(n-1) + \cdots + a_Kr(n-K) = 0, \quad n > 0, r(-n) = r^*(n) \quad (2.32)$$

where $r(n)$ represents the covariance sequence of $\hat{f}(n)$. The probability density function of $f = [\hat{f}(1), \dots, \hat{f}(\tilde{N})]^T$ is given by

$$p_F(f) = \frac{1}{(\pi)^{\tilde{N}} |R_F|} e^{-(f^H R_F^{-1} f)} \quad (2.33)$$

which is also the likelihood function (LF) of f . In (2.33) $[R_F]_{i,j} = E(\hat{f}(i)\hat{f}^*(j)) = r(i-j)$.

Let $a = [a_1, \dots, a_K]^T$. The dependence of LF on a is due to the fact that each element of R_F can be expressed in terms of the AR parameters via the Yule-Walker equations (32). The AR parameters can then be obtained by maximizing the LF by using a non-linear optimization tool. Due to the missing samples, in general R_F will not be Toeplitz and could in fact be singular; apart from that the LF is highly nonlinear in the AR parameters and may have local maxima. Instead of directly maximizing the LF, the authors in [19] have expressed the LF in terms of conditional probability densities. Once the AR parameters were estimated the spectrum of the signal can be obtained as

$$F(\omega) = \left| \frac{\sigma_\varepsilon}{1 + \sum_{k=1}^K a_k e^{-i\omega k}} \right|^2. \quad (2.34)$$

Least squares fitting of AR model to nonuniform samples

As described before, the nonuniform sampling problem can be converted into a missing data problem by a slotted resampler. Following the same notations as in the last section, the AR parameters can be estimated by minimizing the following criterion:

$$\begin{aligned} \min_{\tilde{a}} \sum_{n=K+1}^{\tilde{N}} & |a_0 \hat{f}(n) + a_1 \hat{f}(n-1) + \dots + a_K \hat{f}(n-K)|^2 \\ \text{s.t. } & \tilde{a}^H u = 1 \end{aligned} \quad (2.35)$$

where $\tilde{a} = [a_0, a_1, \dots, a_K]^T$ and $u = [1, 0, \dots, 0]^T$, and the constraint $\tilde{a}^H u = 1$ ensures that the first element of \tilde{a} is equal to one. By introducing a selection matrix

$$M_n = \begin{bmatrix} 0_{n-1} & I_K & 0_{\tilde{N}-K-n+1} \end{bmatrix} \quad (n+1) \times \tilde{N} \quad (2.36)$$

the minimization becomes:

$$\begin{aligned} \min_{\tilde{a}} \tilde{a}^H & \sum_{n=K+1}^{\tilde{N}} (M_n f f^H M_n^T) \tilde{a} \\ \text{s.t. } & \tilde{a}^H u = 1 \end{aligned} \quad (2.37)$$

where $f = [f(1), \dots, f(\tilde{N})]^T$. If all the samples in f are available, then the AR parameters can be obtained analytically as:

$$\tilde{a} = \frac{Q^{-1}u}{u^H Q^{-1}u}. \quad (2.38)$$

where $Q = \sum_{n=K+1}^{\tilde{N}} (M_n f f^H M_n^T)$. Computing Q , however, is not possible if f has some missing samples. Let f_a and f_m represent the available and missing samples in f respectively, such that $f = S_a f_a + S_m f_m$, where S_a and S_m are semiunitary selection matrices such that $f_a = S_a^T f$ and $f_m = S_m^T f$. The cost function in (2.37) is also quadratic in the missing samples as shown below:

$$\begin{aligned} \sum_{n=K+1}^{\tilde{N}} \tilde{a}^H M_n f f^H M_n^T \tilde{a} &= \sum_{n=K+1}^{\tilde{N}} (S_a f_a + S_m f_m)^H M_n^T \tilde{a} \tilde{a}^H M_n (S_a f_a + S_m f_m) \\ &= (S_a f_a + S_m f_m)^H \sum_{n=K+1}^{\tilde{N}} M_n^T \tilde{a} \tilde{a}^H M_n (S_a f_a + S_m f_m) \\ &= (S_a f_a + S_m f_m)^H \tilde{Q} (S_a f_a + S_m f_m) \end{aligned} \quad (2.39)$$

where $\tilde{Q} = \sum_{n=K+1}^{\tilde{N}} M_n^T \tilde{a} \tilde{a}^H M_n$. If the AR parameters in \tilde{Q} were known, then an estimate of f_m could be obtained in the same way as we obtained the estimate

Initialization

Use the estimate (2.38) of \tilde{a} with $f_m = 0$, as the initial value, $\tilde{a}_0 = (u^H Q^{-1} u)^{-1} Q^{-1} u$.

Iteration

At any i^{th} iteration, the estimate of f_m and \tilde{a} is given by:

$$f_m^i = -(S_m^T \tilde{Q}_i^{-1} S_m)^{-1} (S_m^T \tilde{Q}_i^{-1} S_a) f_a$$

$$\tilde{a}_i = (u^H Q_i^{-1} u)^{-1} Q_i^{-1} u$$

where $\tilde{Q}_i = \sum_{n=K+1}^{\tilde{N}} M_n^T \tilde{a}_{i-1} \tilde{a}_{i-1}^H M_n$, $Q_i = \sum_{n=K+1}^{\tilde{N}} (M_n f_i f_i^H M_n^T)$, and $f_i = S_a f_a + S_m f_m^i$.

Termination

The iteration will be terminated when the relative change in \tilde{a} , $\|\tilde{a}_i - \tilde{a}_{i-1}\|^2$, is less than 10^{-4} .

Table 2.5. LS-AR algorithm.

of AR parameters in (2.38). Table 2.5 shows a cyclic iterative algorithm, called LS-AR, that estimates the AR parameters and the missing samples iteratively. In [113], the same sort of cyclic algorithm has been used to estimate an ARX model in the missing data case.

Apart from the above two missing data-based algorithms, there are other missing data algorithms available in the literature. For example the gapped data amplitude and phase estimation (GAPES) algorithm [94] estimates the amplitude spectrum from the available samples, and then estimates the missing samples by a least squares fit to the estimated amplitude spectrum. In the case of arbitrarily missing data, a method called missing data amplitude and phase estimation (MAPES) proposed in [114] estimates the spectrum and the missing samples cyclically via the expectation maximization (EM) algorithm. Somewhat similarly the method called the missing data iterative adaptive approach (MIAA) [97] estimates the spectrum and the missing samples cyclically via a weighted least squares approach.

A numerical simulation has been carried out to compare some of the methods. Two hundred nonuniform samples in the interval $[0 - 10]$ seconds for two sine waves with frequencies 2Hz and 5Hz and corresponding amplitudes 0.5 and 1 are generated with the sampling instances following a stratified uniform distribution as described in section 2.2.2. Using a slotted resampling technique with mean sampling time as the resampling time (i.e. $t_r = t_a$) and half of the mean sampling time as the slot width (i.e. $t_w = \frac{t_g}{2}$), the nonuniform sampling problem has been transformed into a missing data problem. Figure 2.7 shows the spectral estimates of different methods; as can be seen from the figure MIAA gives accurate estimates of frequencies but the estimates of the amplitudes are biased; on the other hand LS-AR gives more accurate estimates of peak heights but poorer resolution than MIAA. When compared with MIAA and LS-AR, the ARFIL fails to locate the peak at 2Hz; the failure of ARFIL is mainly due to the presence of numerous local maxima of the likelihood function due to missing samples.

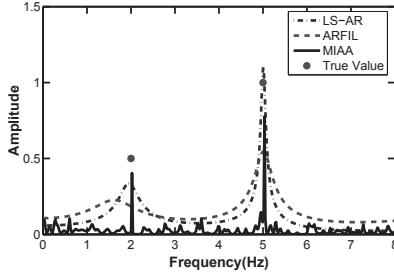


Figure 2.7. Amplitude estimates of the two sine simulated data from the methods based on slotted resampling.

2.4.4 Methods based on continuous time models

In this subsection we describe methods that are based on fitting a continuous time model to nonuniform samples. The methods are based on either approximating the continuous time model by a discrete time model [66] [65] and estimating the parameters of the obtained discrete time model, or approximating the derivative operation in the continuous time model [53] by a weighted summation. As an example, given N nonuniformly spaced samples of $f(t)$, $\{f(t_k)\}_{k=1}^N$, let us briefly describe some methods that try to fit these samples to a CAR(1) model:

$$(p + a)f(t) = e(t) \quad (2.40)$$

where p denotes the differentiation operator, $\frac{d}{dt}$, and $e(t)$ denotes a continuous time white noise of unit intensity. As described in [66] [65], the CAR model can be approximated by the following discrete time model:

$$f(t_n) = e^{-a(t_n - t_{n-1})} f(t_{n-1}) + \sigma_f (1 - e^{2a(t_{n-1} - t_n)})^{1/2} e(t_n) \quad n = 2, \dots, N \quad (2.41)$$

where σ_f^2 denotes the variance of the continuous time process. In [66], a prediction error approach was proposed to estimate a from the model in (2.41).

The above described method is an indirect way of estimating the continuous time model, since it first translates the original model into a discrete time model and then estimates the parameters of the original model by estimating the parameters of the discrete time model. In [53], a more direct approach of estimating the continuous time model parameters has been studied, which approximates the differentiation operator p^j as

$$p^j f(t_k) \approx D^j f(t_k) = \sum_{\mu=0}^j \beta_k(j, \mu) f(t_{k+\mu}) \quad (2.42)$$

where β_k 's are chosen to meet the conditions shown below:

$$\sum_{\mu=0}^j \beta_k(j, \mu) (t_{k+\mu} - t_k)^v = \begin{cases} 0 & v = 0, \dots, j-1 \\ j! & v = j \end{cases} \quad (2.43)$$

Classification				
Sampling type \ Method Type	NP-Discrete spectra	NP-Continuous spectra	P-Discrete spectra	P-Continuous spectra
Missing data case	<ul style="list-style-type: none"> • IAA. • MAPES. • GAPES. 	<ul style="list-style-type: none"> • KDI, KCI + NP uniform sampling methods. • NN + NP uniform sampling methods. 	<ul style="list-style-type: none"> • CLEAN. • KDI, KCI + P uniform sampling methods. 	<ul style="list-style-type: none"> • ML-Jones. • ARFIL. • LS-AR.
Arbitrary irregular sampling	<ul style="list-style-type: none"> • Schuster Periodogram. • LS Periodogram. • IAA. • SR+MIAA. • Sparse approach. • Linear system approach. 			<ul style="list-style-type: none"> • SR + ML-Jones. • SR + ARFIL. • SR + LS-AR. • CAR, CARMA modeling.

Table 2.6. Classification of spectral analysis methods for nonuniform data based on the sampling pattern, signal model and type of spectrum. (P-Parametric, NP-Nonparametric, SR-Slotted Resampling, NN- Nearest Neighbor interpolation, KCI- Kernel based Covariance Interpolation, KDI- Kernel based Data Interpolation)

Using this approximation of the derivative, the CAR model in (2.40) can be rewritten as

$$D^1 f(t_k) = -a D^0 f(t_k) + e(t_k) \quad k = 1, \dots, N-1. \quad (2.44)$$

From the above set of equations the parameter a can be obtained by least squares. The interested reader is referred to [53] [65] [66] and the references there for technical details and numerical results on fitting continuous time models to nonuniform samples.

2.4.5 Classification and summary

Table 2.6 shows a classification of different spectral analysis methods based on the sampling pattern, signal model and type of spectrum. Some of the methods listed in the table are new. For example, both MIAA and LS-AR were previously applied only on missing data case, but here they are used in conjunction with slotted resampling on arbitrarily sampled data. Other methods like the sparse approach and the linear system approach, which are listed under exact signal reconstruction methods, can also be used for spectral analysis of finite-length nonuniform data.

2.4.6 Performance on real life data sets

In this subsection, the performance of some of the above methods on two real life data sets is evaluated. For each data set, the following results are provided:

- The Nyquist frequency for the sampling pattern, obtained from the spectral window.
- The spectral estimates from applicable methods (according to the sampling pattern type). For example, in the both real life examples the sampling pattern is arbitrary and only the following methods can be applied:
 - methods based on least squares.
 - missing data methods applied to slotted resampled data.
 - uniform sampling methods like periodogram, Capon (nonparametric) and ESPRIT (parametric), which are applied to the interpolated covariance sequence.

The methods of sparse approach and of linear system approach are not applied here since the real life data sets are neither sparse nor periodic bandlimited. Regarding the IAA based methods, three types of methods are possible: IAA applied directly to nonuniform data; IAA applied to slotted resampled data; MIAA applied to slotted resampled data. Out of them MIAA applied to slotted resampled data is preferred over applying IAA on slotted resampled data, as the later is the first step of the iteration in the former.

Radial velocity data

The data considered express the radial velocity of the star HD 102195. Based on these data, an extrasolar planet named ET-1 revolving around the star has been discovered [35]. The period of revolution of the planet can be calculated by analyzing the spectrum of the radial velocity measurements. The experimental data consists of 38 sample measurements of the radial velocity of the star taken nonuniformly over a span of 200 days as shown in the figure 2.8. Figure 2.9 shows the sampling pattern and the spectral window of the data; as can be seen from the spectral window the spectrum replicates after a frequency of about 1 cycles/day. Table 2.7 shows some spectral parameters of the radial velocity data; it can be seen from the table that although the rollover frequency of the spectrum calculated from t_m is much larger than that calculated from t_r , the Nyquist frequency is still taken to be Ω_{nus} . Figures 2.10 and 2.11 show the spectral estimates of the radial velocity for the LS-methods, as well as methods based on slotted resampling, and covariance interpolation methods respectively. In the case of LS based methods, the LS-AR method cannot be applied to this data due to the big gaps in the sampling pattern. Table 2.8 summarizes the results obtained from the different methods. Excepting the ARFIL method, all the methods indicate a strong periodicity around 4.1 days (0.2439 cycles/day); so the period of rotation of the exoplanet, ET-1, around the star HD 102195 has been estimated to be 4.1 days.

Spectral parameter	Value
Average sampling time (t_a) and $\Omega_a/2\pi$	5.2266 days, 0.0957 cycles/day
Minimum time interval (t_m) and $\Omega_m/2\pi$	0.0078 days, 64.0533 cycles/day
Resampling time (t_r) and $\Omega_{rus}/2\pi$	0.5 days, 0.5 cycles/day

Table 2.7. Spectral parameters of the radial velocity data.

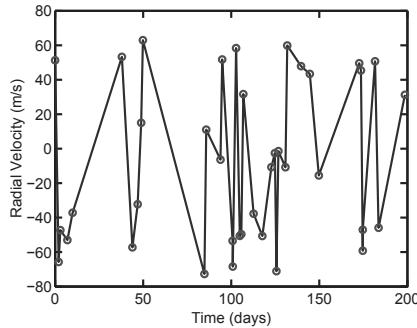
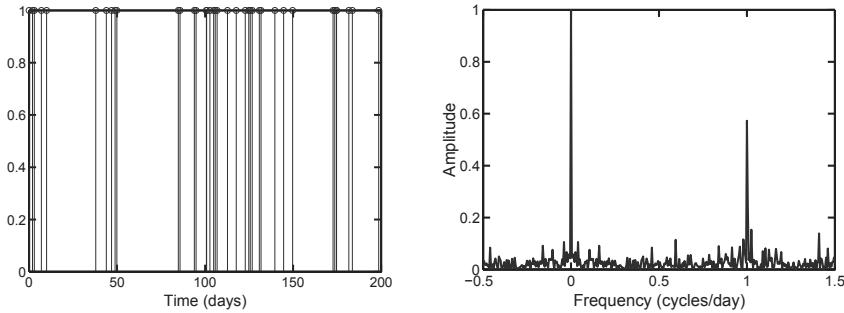


Figure 2.8. The radial velocity data of HD 102195.



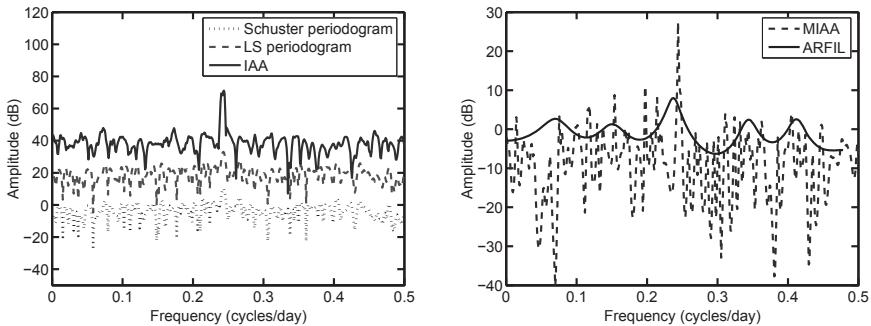
(a) Sampling pattern of the data.

(b) Spectral window.

Figure 2.9. The sampling pattern and the spectral window of the radial velocity data.

Type	Method	Cycles/day
LS methods	Schuster periodogram	0.2439
	LS periodogram	0.2439
	IAA	0.2439
Methods based on slotted resampling	MIAA	0.2447
	ARFIL	0.23
Covariance interpolation (Sinc kernel) + Non parametric methods	Periodogram	0.2439
	Capon	0.2439
Covariance interpolation (Sinc kernel) + Parametric method	ESPRIT	0.2431

Table 2.8. The main spectral peak for the radial velocity data selected by different methods.



(a) LS methods.

(b) Methods based on slotted resampling.

Figure 2.10. Spectrum of the radial velocity data based on a) LS methods (the scales in the plot are changed to show the spectra from different methods clearly), b) Methods based on slotted resampling (for MIAA and ARFIL, the spacing between the samples is taken to be equal to t_r).

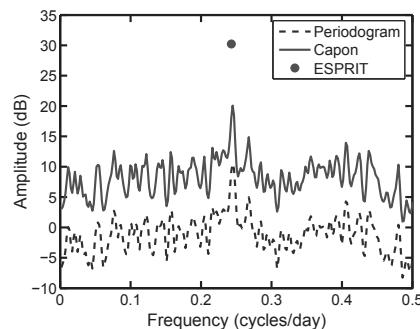


Figure 2.11. Spectrum of the radial velocity data obtained by covariance interpolation using a sinc kernel with mainlobe width $b_1 = t_r$ and then applying two nonparametric uniform sampling methods (Periodogram and Capon) and a parametric uniform sampling method (ESPRIT).

Spectral parameter	Value
Average sampling time (t_a) and $\Omega_a/2\pi$	1.4511×10^3 years, 3.4457×10^{-4} cycles/year
Minimum time interval (t_m) and $\Omega_m/2\pi$	44 years, 0.0114 cycles/year
Resampling time (t_r) and $\Omega_{rus}/2\pi$	0.5 years, 0.5 cycles/year

Table 2.9. Spectral parameters of the Vostok ice core data.

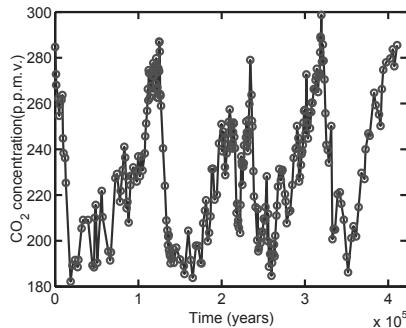
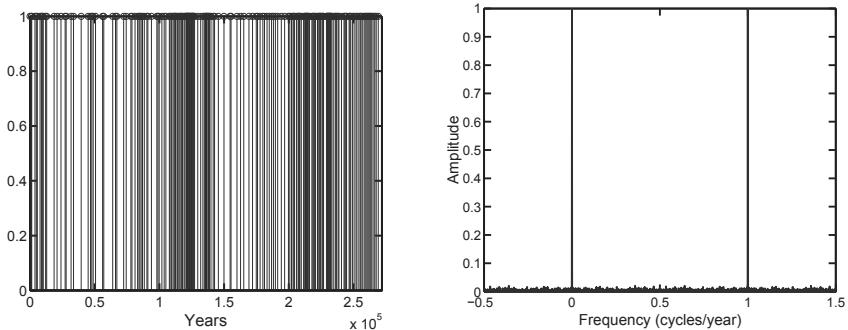


Figure 2.12. Vostok ice core data.

Vostok ice core data

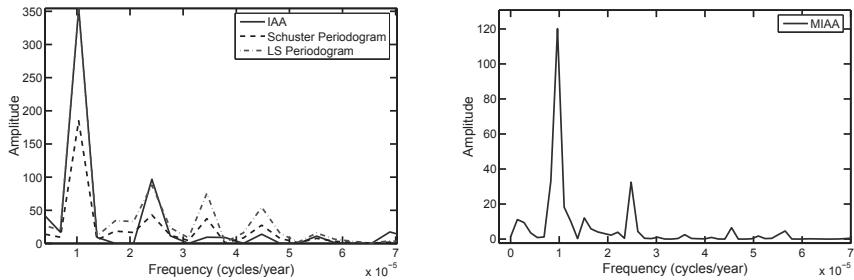
The data used in this example are concentrations of CO_2 measured in parts per million by volume (p.p.m.v.), obtained from ice core drilling at the Russian Vostok station in East Antarctica [77] [69]. Analyzing these data will be helpful in knowing the paleoclimatic properties of the glacial-interglacial periods such as temperature, wind speed, changes in atmospheric gas composition, etc. The data consists of 283 samples of CO_2 concentration ranging over a span of 420,000 years as shown in the figure 2.12. Figure 2.13 shows the sampling pattern and its spectral window; as can be seen from the spectral window the Nyquist frequency is around 0.5 cycles/year. Table 2.9 shows some spectral parameters of the Vostok ice core data. In the case of methods based on slotted resampling, the results of LS-AR and ARFIL methods are not shown; LS-AR cannot be applied here due to large gaps and ARFIL fails to locate any peak in the spectrum. Figures 2.14 and 2.15 show the spectral estimates of the Vostok ice core data for the LS-methods, methods based on slotted resampling, and covariance interpolation methods, respectively. Since the spectra obtained from all the methods do not show any periodicity beyond 7×10^{-5} cycles/year, the spectra are shown only till 7×10^{-5} cycles/year. The mainlobe width of the sinc kernel in the case of covariance interpolation techniques is chosen somewhat arbitrarily to be $3t_a$. Table 2.10 summarizes the results obtained from the different methods. All the methods indicate a strong periodicity around 10^5 years (1×10^{-5} cycles/year), and relatively weaker periodicities at 40,000 years (2.5×10^{-5} cycles/year), 28,571 years (3.5×10^{-5} cycles/year) and 22,222 years (4.5×10^{-5} cycles/year).



(a) Sampling pattern.

(b) Spectral window.

Figure 2.13. The sampling pattern and the spectral window of the Vostok ice core data.



(a) LS methods.

(b) Methods based on slotted resampling.

Figure 2.14. Spectrum of the Vostok ice core data based on a) LS methods, b) Methods based on slotted resampling (for MIAA, the spacing between samples is taken to be equal to t_r).

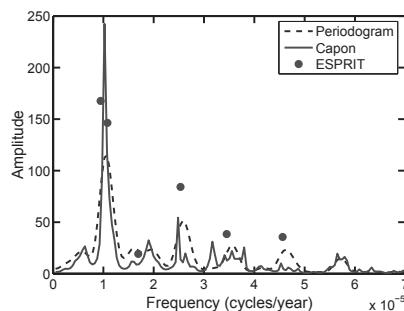


Figure 2.15. Spectrum of the Vostok ice core data obtained by covariance interpolation using a sinc kernel with mainlobe width $b_1 = 3t_a$ and applying two nonparametric uniform sampling methods (Periodogram and Capon) and a parametric uniform sampling method (ESPRIT).

Type	Method	Cycles/year
LS methods	Schuster periodogram	$\{1.034, 2.412\} \times 10^{-5}$
	LS periodogram	$\{1.034, 2.412\} \times 10^{-5}$
	IAA	$\{1.034, 2.412\} \times 10^{-5}$
Methods based on slotted resampling	MIAA	$\{0.9648, 2.481\} \times 10^{-5}$
Covariance interpolation (Sinc kernel)+ Non parametric methods	Periodogram Capon	$\{1.071, 2.581\} \times 10^{-5}$ $\{1.023, 2.484\} \times 10^{-5}$
Covariance interpolation (Sinc kernel)+ Parametric method	ESPRIT	$\{0.9437, 1.08, 2.53\} \times 10^{-5}$

Table 2.10. The two most significant spectral peaks (three peaks for ESPRIT) in the Vostok ice core data picked by different methods.

2.5 Conclusions

In this chapter, we have reviewed different methods for spectral analysis of nonuniform data by describing, classifying and comparing them. Apart from methods for spectral analysis, methods for exact signal reconstruction from nonuniform samples were also reviewed. The choice of various spectral parameters like Nyquist frequency, and resampling rate were also discussed. Finally the performance of different spectral analysis methods on two real-life nonuniform data sets, one in astrophysics and the other in paleoclimatology, was evaluated. For both real life data sets it was observed that the nonparametric methods perform better than the parametric methods, especially the IAA based methods showed clearly periodicities in the data which agree well with physical findings.

3. A nonparametric approach to estimation of smooth spectra from nonuniformly sampled data

3.1 Introduction

Non-parametric estimation of line (impulse-like) spectra from irregularly sampled data has received significant attention in the recent years and several methods have been proposed (see, e.g., [122] [96] [93] [34] and the references there) that significantly outperform the classical periodogram (e.g. [86] [59]). Specifically, these newly introduced methods, which yield either dense or more commonly sparse spectral estimates, have higher resolution and much smaller leakage problems than the periodogram.

The same cannot be said about the estimation of continuous spectra from unevenly sampled data, a topic that appears to be underdeveloped at present. Indeed the methods mentioned in the previous paragraph are not usable in the continuous spectrum case. Furthermore, most of the existing methods for tackling the estimation of such spectra are based on heuristic interpolation ideas that may introduce unnecessary approximations in their attempt to transform the problem into an evenly sampled data one (see, e.g., [100] [6] [20] [21] [83] [68] and the many references therein). One of the few methods that do not recourse to interpolation is **D**Aniell Method (DAM) which obtains spectral estimates by locally averaging the periodogram under the assumption that the spectrum is smooth. The statistical performance of DAM, however, is usually rather low in the irregular-data case in which the periodogram values may be strongly correlated with one another and therefore averaging them may not be an effective means of reducing their large variance. Nevertheless, DAM is a computationally simple method that constitutes an obvious competitor for any new spectral estimation method and which therefore will be considered here for comparison purposes.

In this chapter we will introduce a method for non-parametric spectral estimation based on the maximum-likelihood (ML) principle, which will be designated by the acronym LIMES (**L**Ikelihood-based **M**ethod for **E**stimation of **S**pectra). LIMES operates under the following assumptions :

- A1) The spectrum is band-limited to a given frequency B ;
- A2) The sampling pattern, while virtually arbitrary, should allow alias-free spectral estimation up to a maximum frequency (the so-called folding frequency) that is larger than B ; and

A3) The spectrum is piecewise linear (or nearly so) over given small frequency bins (to be defined shortly).

These assumptions are quite practical. Indeed A1 is a necessary assumption for any method that, like LIMES, estimates all spectral components jointly (as opposed to DAM which estimates them one by one and therefore does not impose A1; yet DAM's performance depends heavily on A1, see Section 3.7). Furthermore, A2 is a natural requirement without which a spectral estimation method can hardly produce meaningful results. Finally, A3 is a weaker assumption than DAM's related requirement that the spectrum is piecewise constant.

In the next section we will mathematically formulate the spectral estimation problem of interest and show how to check that A2 is satisfied. DAM will also be briefly reviewed in this section. Section 3.3 makes use of A1 and A3 to derive a non-parametric representation of the data covariance matrix (which can be viewed as an extension of the covariance model in [103] that was based on the piecewise-constant spectrum assumption). The representation of Section 3.3 lies at the basis of the ML estimation approach proposed in Section 3.4. The latter section also presents the Cramer-Rao bound (CRB) associated with the estimation problem under consideration. In Section 3.5 and 3.6 we derive the LIMES algorithm for computing the ML spectral estimate introduced in the previous section. Finally in Section 3.7 we present several numerical examples, including an application to real-life data, to illustrate the statistical performance of LIMES and compare it with the performance of DAM and with the CRB.

3.2 Problem formulation and DAM

Let $y(t) \in \mathbb{C}$, for $t \in \mathbb{R}$, be a zero-mean stationary continuous-time signal and let $r(\tau)$ denote its covariance function :

$$r(\tau) = E [y(t)y^*(t - \tau)], \tau \in \mathbb{R} \quad (3.1)$$

(hereafter, $*$ denotes the complex conjugate for scalars and the conjugate transpose for matrices). The power spectral density, or the spectrum for short, is then given (under quite general conditions on $r(\tau)$, see e.g. [78] [17]) by:

$$\phi(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} r(\tau) e^{-i\omega\tau} d\tau \quad (3.2)$$

where $\omega \in \mathbb{R}$ is the frequency variable (scaled by 2π). Under A1 the spectrum is band-limited:

$$\phi(\omega) = 0 \text{ for } |\omega| > B \quad (3.3)$$

The main problem of interest in this chapter is the estimation of $\phi(\omega)$, for $\omega \in [-B, B]$, from a set of N in general irregularly sampled observations of

$y(t)$:

$$y = [y(t_1), \dots, y(t_N)]^T \quad (3.4)$$

In the course of estimating $\phi(\omega)$ we will also obtain, as a byproduct, an estimate of the covariance matrix of y :

$$R = E[yy^*] = \int_{-B}^B \phi(\omega) a(\omega) a^*(\omega) d\omega \quad (3.5)$$

where

$$a(\omega) = [e^{i\omega t_1}, \dots, e^{i\omega t_N}]^T \quad (3.6)$$

Note that, for the sake of generality, we have assumed that $y(t) \in \mathbb{C}$. A real-valued signal $y(t) \in \mathbb{R}$ can be treated in our framework as a special case of complex-valued signal for which the following symmetry condition holds:

$$\phi(-\omega) = \phi(\omega), \quad \omega \in [-B, B] \quad (3.7)$$

Because this condition is easily imposed in the proposed approach (see the next sections) we will omit any further discussion on it in what follows.

Regarding the observation times $\{t_n\}$, while they can be quite arbitrary (as already indicated above), they of course have to satisfy the alias-free sampling condition embodied in A2. This condition requires that the vectors $a(\omega)$ and $a(\tilde{\omega})$ (with $\omega \neq \tilde{\omega}$) should not be (quasi) parallel for $|\omega - \tilde{\omega}| \leq 2B$. Letting $v = \omega - \tilde{\omega}$ we can therefore formulate A2 mathematically as:

$$S(v) \stackrel{\Delta}{=} \left| \sum_{n=1}^N e^{ivt_n} \right| << S(0) = N \text{ for } |v| \leq 2B \quad (3.8)$$

where $S(v)$ is sometimes called the spectral window. Many random sampling patterns encountered in applications satisfy (3.8) for quite large values of B . In particular if $\min_n [t_{n+1} - t_n]$ is on the order of 10^{-d} [sec], for some $d \geq 1$, then the smallest value of $v \neq 0$ that can lead to $S(v) \approx N$ is typically on the order of 10^d [rad/sec] (see e.g. [31]). Satisfying (3.8) should therefore be no serious problem in general if d is or can be chosen to be relatively large.

The third assumption made above can be stated mathematically as explained next. Let

$$\Delta = 2B/M \quad (3.9)$$

where M is a given integer. Then A3 states that $\phi(\omega)$ varies linearly over consecutive frequency bins of size Δ :

$$\phi(\omega) = \frac{\omega - \omega_k}{\Delta} \phi_{k+1} + \frac{\omega_{k+1} - \omega}{\Delta} \phi_k \quad \text{for } \omega \in [\omega_k, \omega_{k+1}] \quad (3.10)$$

and $k = 1, \dots, M$

where

$$\omega_k = -B + (k-1)\Delta \quad k = 1, \dots, M+1 \quad (3.11)$$

and

$$\phi_k = \phi(\omega_k) \quad k = 1, \dots, M+1 \quad (3.12)$$

Making use of this assumption we can therefore reduce the problem of estimating the entire spectrum $\phi(\omega)$ to that of estimating $M+1$ unknowns $\{\phi_k\}_{k=1}^{M+1}$ from $\{y(t_n)\}_{n=1}^N$. The latter problem should be more manageable than the former as long as M is smaller than N . The choice of M should be done with the usual interplay between bias and variance in mind. Specifically, the estimates of $\{\phi_k\}$ from the N available data samples can be expected to have small variance if $M \ll N$. On the other hand, equation (3.10) may fail to hold with a reasonable accuracy unless $M \gg 1$; in other words, the estimation bias, induced by the approximate character of (3.10), may be large if M is chosen too small. In general the selection of M should be based on prior knowledge : the smoother the spectrum is expected to be the smaller M can be chosen. Roughly speaking, a value of $M \geq 30$ should be sufficient for (3.10) to approximate well a reasonably smooth spectrum and hence for keeping the bias small. At the same time, we would also like to keep the estimation variance small, which may require (once again, roughly speaking) that $N \geq 10M$. For $M = 30$, e.g., this sets a limit on the minimum acceptable number of data samples : $N = 300$. Put differently, for a given N we can estimate reliably (i.e. with small variance and bias) only spectra that are smooth enough to be well approximated by (3.10) with, let us say, $M \leq N/10$.

We end this section with a brief review of DAM. Let

$$\mu_k = -B + (k-1) \frac{2B}{N} \quad k = 1, \dots, N+1 \quad (3.13)$$

and let

$$m = \lfloor N/M \rfloor \quad (3.14)$$

(i.e. the integer part of N/M). Then, assuming that the spectrum is (nearly) constant over intervals of length $2Bm/N$, DAM obtains a spectral estimate by averaging the values of the periodogram at the current frequency and at the m frequencies around it ([28] [98]) :

$$\phi(\mu_k) = \frac{1}{2BN(m+1)} \sum_{p=\max(1,k-m/2)}^{\min(N+1,k+m/2)} |a^*(\mu_p)y|^2 \quad k = 1, \dots, N+1 \quad (3.15)$$

(above we have assumed that m is even, for simplicity; if m is odd then we can use $m+1$ in lieu of m in (3.15)). The scaling factor that appears in front of the sum in (3.15) can be motivated as follows: $|a^*(\mu_p)y|^2/N^2$ is the periodogram estimate of the power in the p -th frequency bin whose width is equal to $2B/N$; therefore the value of the spectrum at μ_p can be estimated as $\frac{N}{2B} |a^*(\mu_p)y|^2/N^2 = |a^*(\mu_p)y|^2/2BN$, and these are the values averaged in (3.15).

3.3 Non-parametric model of R

The (p,s) -element of R has the following expression (see A1 and (3.5)) :

$$R_{ps} = \int_{-B}^B \phi(\omega) e^{i\omega\tau} d\omega \quad (3.16)$$

where $\tau = t_p - t_s$ (we omit the indexes p and s of τ for notational simplicity). Making use of (3.16) along with A3 and (3.10) we obtain :

$$R_{ps} = \sum_{k=1}^M \int_{\omega_k}^{\omega_{k+1}} \phi(\omega) e^{i\omega\tau} d\omega = \sum_{k=1}^M [\phi_{k+1} F_{ps}(k+1) + \phi_k G_{ps}(k)] \quad (3.17)$$

where

$$F_{ps}(k+1) = \frac{1}{\Delta} \int_{\omega_k}^{\omega_{k+1}} (\omega - \omega_k) e^{i\omega\tau} d\omega \quad (3.18)$$

$$G_{ps}(k) = -\frac{1}{\Delta} \int_{\omega_k}^{\omega_{k+1}} (\omega - \omega_{k+1}) e^{i\omega\tau} d\omega \quad (3.19)$$

The above integrals are evaluated in section 7.5. It follows from the expressions for $F_{ps}(k+1)$ and $G_{ps}(k)$ derived there that :

$$F(k+1) = D_{k+1} \Pi D_{k+1}^* \quad (3.20)$$

$$G(k) = D_k \Gamma D_k^* \quad (3.21)$$

where D_k is the diagonal modulation matrix for the k -th frequency bin, i.e.

$$D_k = \begin{bmatrix} e^{i\omega_k t_1} & 0 & \dots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \dots & \ddots & \vdots \\ 0 & \dots & \dots & e^{i\omega_k t_N} \end{bmatrix} \quad (3.22)$$

and Π and Γ are positive semi-definite matrices with the following elements

$$\Pi_{ps} = \begin{cases} \frac{1}{i(t_p - t_s)} + \frac{1}{\Delta(t_p - t_s)^2} [1 - e^{-i\Delta(t_p - t_s)}] & p \neq s \\ \Delta/2 & p = s \end{cases} \quad (3.23)$$

$$\Gamma_{ps} = \begin{cases} -\frac{1}{i(t_p - t_s)} + \frac{1}{\Delta(t_p - t_s)^2} [1 - e^{i\Delta(t_p - t_s)}] & p \neq s \\ \Delta/2 & p = s \end{cases} \quad (3.24)$$

Inserting (3.20) and (3.21) in (3.17) yields the following expression for R :

$$R = \phi_1 G(1) + \sum_{k=2}^M \phi_k [F(k) + G(k)] + \phi_{M+1} F(M+1) \quad (3.25)$$

where

$$F(k) + G(k) = D_k (\Pi + \Gamma) D_k^* \quad (3.26)$$

In (3.26), $\Pi + \Gamma$ is a real-valued matrix with

$$(\Pi + \Gamma)_{ps} = \Delta \quad (p = s) \quad (3.27)$$

and with the following off-diagonal elements

$$\begin{aligned} (\Pi + \Gamma)_{ps} &= \frac{2}{\Delta(t_p - t_s)^2} [1 - \cos(\Delta(t_p - t_s))] \\ &= \frac{4}{\Delta(t_p - t_s)^2} [\sin(\Delta(t_p - t_s)/2)]^2 = \Delta [\text{sinc}(\Delta(t_p - t_s)/2)]^2 \quad (p \neq s) \end{aligned} \quad (3.28)$$

where $\text{sinc}(z) = \sin(z)/z$.

Remark 1 For real-valued signals, for which the symmetry condition $\phi_k = \phi_{M+2-k}$ ($k = 1, \dots, M+1$) holds true, the right-hand side of (3.25) is a real-valued matrix as it should.

Equation (3.25) constitutes the non-parametric model for R that will be used in this chapter. By letting

$$C_k = \begin{cases} G(1) & k = 1 \\ F(k) + G(k) & k = 2, \dots, M \\ F(M+1) & k = M+1 \end{cases} \quad (3.29)$$

we can write (3.25) in the following more compact form :

$$R = \sum_{k=1}^{M+1} \phi_k C_k \quad (3.30)$$

Depending on Δ and the sampling pattern $\{t_n\}$, the rank of the positive semi-definite matrices $\{C_k\}$ may be less than N . Let

$$m_k = \text{rank}(C_k) \quad (3.31)$$

be the effective rank of C_k as determined, for example, by the Matlab function **rank**. Also, let $A_k \in \mathbb{C}^{N \times m_k}$ be a rank- m_k square root of C_k , i.e.

$$C_k = A_k A_k^* \quad (3.32)$$

Then we can write (3.30), with a good approximation, in the following factorized form:

$$R = \sum_{k=1}^{M+1} \phi_k A_k A_k^* = APA^* \quad (3.33)$$

where

$$A = [A_1, \dots, A_{M+1}] \quad (3.34)$$

and

$$P = \begin{bmatrix} \phi_1 I_{m_1} & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \phi_{M+1} I_{m_{M+1}} \end{bmatrix} \quad (3.35)$$

Note that

$$\text{rank}(C_k) = \text{rank}(\Pi + \Gamma) \stackrel{\Delta}{=} \rho \quad k = 2, \dots, M \quad (3.36)$$

and

$$A_k = D_k S \quad k = 2, \dots, M \quad (3.37)$$

where $S \in \mathbb{C}^{N \times \rho}$ is a square-root of $\Pi + \Gamma$. Consequently the computation of $\{A_k\}_{k=1}^{M+1}$ requires the rank determination and square-root factorization of only three matrices and therefore it does not add much to the computational burden of LIMES (see the next sections). In fact the total computational burden of the algorithm may well decrease owing to the above reduction of the dimension of A_k from $N \times N$ to $N \times m_k$ ($k = 1, \dots, M+1$).

3.4 ML approach and the CRB

If the data vector y is assumed to have a circular normal distribution with zero mean and covariance matrix R , then the corresponding negative log-likelihood function is given (to within an additive constant) by :

$$f(\phi) = \ln|R| + y^* R^{-1} y \quad (3.38)$$

where $|R|$ denotes the determinant of R ,

$$\phi = [\phi_1, \dots, \phi_{M+1}]^T \quad (\phi_k \geq 0) \quad (3.39)$$

and (see (3.33))

$$R = APA^* \quad (3.40)$$

is assumed to be nonsingular (which is a weak condition, in general). Note that (3.38) is a well-established data fitting criterion even if the observations are *not* normally distributed.

We will obtain spectral estimates by minimizing (3.38) with respect to ϕ . While this minimization problem is non-convex (see below) and therefore potentially difficult to solve globally, we will derive an algorithm for solving it that decreases $f(\phi)$ at each iteration and which can thus be expected to converge at least locally (see the next sections).

The first term in (3.38) is a *concave function* of ϕ . This follows from the fact that the Hessian matrix associated with this term is negative semi-definite

at any point in the parameter space. To prove this claim, first observe that :

$$\begin{aligned}\frac{\partial^2 \ln|R|}{\partial \phi_s \partial \phi_k} &= \frac{\partial \left[\text{tr} \left(R^{-1} \frac{\partial R}{\partial \phi_k} \right) \right]}{\partial \phi_s} = \frac{\partial \left[\text{tr} \left(R^{-1} C_k \right) \right]}{\partial \phi_s} \\ &= -\text{tr}(R^{-1} C_s R^{-1} C_k)\end{aligned}\quad (3.41)$$

where tr is the trace operator. Let

$$H_{sk} = \text{tr}(R^{-1} C_s R^{-1} C_k) \quad (3.42)$$

The stated convexity property follows if one can show that $g^* H g \geq 0$ for any vector $g = [g_1, \dots, g_{M+1}]^T$. A simple calculation yields :

$$g^* H g = \sum_{s=1}^{M+1} \sum_{k=1}^{M+1} g_s^* g_k \text{tr}(R^{-1} C_s R^{-1} C_k) = \text{tr}(R^{-1} X^* R^{-1} X) \quad (3.43)$$

where

$$X = \sum_{k=1}^{M+1} g_k C_k \quad (3.44)$$

and

$$X^* = \sum_{s=1}^{M+1} g_s^* C_s^* = \sum_{s=1}^{M+1} g_s^* C_s \quad (3.45)$$

Next we note that

$$\begin{aligned}\text{tr}(R^{-1} X^* R^{-1} X) &= \text{vec}^*(X) \text{vec}(R^{-1} X R^{-1}) \\ &= \text{vec}^*(X) (R^{-T} \otimes R^{-1}) \text{vec}(X)\end{aligned}\quad (3.46)$$

where \otimes denotes the Kronecker matrix product and $\text{vec}(X)$ is the vector obtained by stacking the columns of X on top of each other. Because the matrix $R^{-T} \otimes R^{-1}$ appearing in (3.46) is positive definite, the right-hand side of this equation cannot be negative, and thus the proof is concluded.

The second term in (3.38) is a *convex function* of ϕ . Indeed, the Hessian matrix corresponding to this term, viz.

$$\begin{aligned}\frac{\partial^2 (y^* R^{-1} y)}{\partial \phi_s \partial \phi_k} &= -\frac{\partial (y^* R^{-1} C_k R^{-1} y)}{\partial \phi_s} = 2\text{Re}(y^* R^{-1} C_s R^{-1} C_k R^{-1} y) \\ &= 2\text{Re}(\text{tr}[(R^{-1} y y^* R^{-1}) C_s R^{-1} C_k])\end{aligned}\quad (3.47)$$

(here $\text{Re}(z)$ denotes the real part of z), can be shown to be positive semi-definite by a calculation similar to (3.43)-(3.46) above.

The non-convexity of $f(\phi)$ follows essentially from the above calculations. In particular note that if $\|\phi\|$ is sufficiently large ($\|\cdot\|$ denotes the Euclidean vector norm) then the negative semi-definite matrix (3.41) dominates (3.47) in the Hessian matrix of $f(\phi)$ and thus the latter matrix cannot be positive semi-definite at all points in the parameter space as would be required for $f(\phi)$ to be convex.

The LIMES algorithm for the minimization of $f(\phi)$ will be derived in the next two sections. In the rest of this section we present the CRB associated with the estimation problem under discussion. Making use of Bangs formula ([8] [98]) it follows that the CRB for the present estimation problem is given by:

$$\text{CRB} = H^{-1} \quad (3.48)$$

where H is the matrix defined in (3.42), which is assumed to be nonsingular. It is an immediate consequence of the calculation which led to (3.46) that H^{-1} exists if and only if

$$X = 0 \iff g = 0 \quad (3.49)$$

Because

$$\text{vec}(X) = \sum_{k=1}^{M+1} g_k \text{vec}(C_k) \quad (3.50)$$

it follows that (3.49) is equivalent to :

$$\text{rank}([\text{vec}(C_1), \dots, \text{vec}(C_{M+1})]) = M+1 \quad (3.51)$$

The matrix in the above equation has many more rows than columns (N^2 vs. $M+1$) and the condition (3.51) for CRB to exist is usually satisfied. Note that this condition can be readily checked in any given case. We also note that the ML estimation of ϕ can be expected to achieve the CRB as both N and N/M increase. Finally we remark in passing that whenever (3.51) holds the first term of $f(\phi)$ is a *strictly* concave function of ϕ .

3.5 LIMES - the basic idea

We will derive an algorithm for minimizing the negative log-likelihood function $f(\phi)$ in (3.38) by using the minimization-majorization (MM) principle (see, e.g., [101] and some of the references therein). To explain how this can be done, let us consider an arbitrary point $\tilde{\phi}$ in the parameter space and assume that a function $h(\phi)$ is available such that:

$$h(\tilde{\phi}) = f(\tilde{\phi}) \quad (3.52)$$

and

$$h(\phi) \geq f(\phi) \quad \forall \phi \quad (3.53)$$

Furthermore, let $\hat{\phi}$ be either the minimum point of $h(\phi)$ or at least a point with the property that

$$h(\hat{\phi}) < h(\tilde{\phi}) \quad (3.54)$$

Then making use of (3.53), (3.54) and (3.52) (in this order) we see that

$$f(\hat{\phi}) \leq h(\hat{\phi}) < h(\tilde{\phi}) = f(\tilde{\phi}) \quad (3.55)$$

and therefore that $f(\phi)$ takes on a smaller value at $\phi = \hat{\phi}$ than at $\phi = \tilde{\phi}$:

$$f(\hat{\phi}) < f(\tilde{\phi}) \quad (3.56)$$

The usefulness of the outlined MM approach to decreasing $f(\phi)$ from $f(\tilde{\phi})$ to $f(\hat{\phi})$ and so forth evidently depends on whether $h(\phi)$ is easier to minimize (or decrease) than $f(\phi)$ is. In the present case, a convenient majorizing function $h(\phi)$ that is *convex* can be found by exploiting the concavity of the first term of $f(\phi)$, as explained next.

A concave function is majorized by its tangent plane at any point $\tilde{\phi}$. Consequently (hereafter, \tilde{R} corresponds to $\tilde{\phi}$) :

$$\begin{aligned} \ln |R| &\leq \ln |\tilde{R}| + \sum_{k=1}^{M+1} \text{tr}(\tilde{R}^{-1} C_k)(\phi_k - \tilde{\phi}_k) \\ &= \ln |\tilde{R}| - N + \text{tr}(\tilde{R}^{-1} R) \end{aligned} \quad (3.57)$$

from which it follows that

$$f(\phi) \leq [\ln |\tilde{R}| - N] + \text{tr}(\tilde{R}^{-1} R) + y^* R^{-1} y \stackrel{\Delta}{=} h(\phi) \quad (3.58)$$

(for completeness sake, we include a direct algebraic proof of (3.57) in 7.6).

The minimization of $h(\phi)$ above, to compute $\hat{\phi}$, is equivalent to :

$$\min_{\phi} \sum_{k=1}^{M+1} \tilde{w}_k \phi_k + y^* R^{-1} y \quad (3.59)$$

where the weights $\{\tilde{w}_k > 0\}$ are given by

$$\tilde{w}_k = \text{tr}(\tilde{R}^{-1} C_k) \quad k = 1, \dots, M+1 \quad (3.60)$$

This minimization problem can be re-formulated as a Semi-Definite Program (SDP), viz. (with α being an auxiliary variable)

$$\begin{aligned} \min_{\alpha, \phi} & \alpha + \sum_{k=1}^{M+1} \tilde{w}_k \phi_k \\ \text{s.t. } & \begin{bmatrix} \alpha & y^* \\ y & R \end{bmatrix} \geq 0 \quad R = A P A^* \end{aligned} \quad (3.61)$$

and is therefore *convex*. More interestingly, the solution of (3.59) can also be derived from that of a Second-Order Cone Program (SOCP), which can be solved much more efficiently than the above SDP, see the next section for details.

3.6 LIMES - the algorithm

To explain how the solution of (3.59) can be obtained by solving an SOCP, instead of the more computationally complex SDP in (3.61), consider the fol-

lowing augmented problem :

$$\begin{aligned} & \min_{\beta, \phi} \beta^* P^{-1} \beta + \sum_{k=1}^{M+1} \tilde{w}_k \phi_k \\ & \text{s.t. } A\beta = y \end{aligned} \quad (3.62)$$

where $\beta \in \mathbb{C}^L$ (with $L = \sum_{k=1}^{M+1} m_k$) is a vector of auxiliary variables. Note that, in order to simplify the following discussion, we have assumed that $\phi_k > 0$ (for $k = 1, \dots, M+1$) such that P^{-1} in (3.62) exists; the more general case of $\{\phi_k \geq 0\}$ can be similarly treated after replacing P^{-1} in the above equation by the pseudo-inverse of P .

To establish the connection between the problems (3.59) and (3.62), we note that the (constrained) minimization of the objective in (3.62) with respect to β , for fixed ϕ , yields :

$$\hat{\beta} = PA^* R^{-1} y \quad (3.63)$$

and

$$\hat{\beta}^* P^{-1} \hat{\beta} + \sum_{k=1}^{M+1} \tilde{w}_k \phi_k = y^* R^{-1} y + \sum_{k=1}^{M+1} \tilde{w}_k \phi_k \quad (3.64)$$

(The proof of this claim can be found in 7.7). The right-hand side of (3.64) is nothing but the objective function of the problem $\min_{\phi} h(\phi)$, see (3.59). It follows that (3.59) and (3.62) have the same minimizer ϕ . Interestingly, it is easier to find this minimizer by solving (3.62) than from (3.59), despite the fact that (3.62) comprises L additional variables, see below.

Let

$$\beta = \left[\beta_1^T, \dots, \beta_{M+1}^T \right]^T ; \beta_k \in \mathbb{C}^{m_k} (k = 1, \dots, M+1) \quad (3.65)$$

The minimization of (3.62) with respect to ϕ , for fixed β , decouples in the following one-dimensional problems :

$$\min_{\phi_k} \|\beta_k\|^2 / \phi_k + \tilde{w}_k \phi_k \quad k = 1, \dots, M+1 \quad (3.66)$$

The solution to (3.66) is readily shown to be:

$$\hat{\phi}_k = \|\beta_k\| / \tilde{w}_k^{1/2} \quad k = 1, \dots, M+1 \quad (3.67)$$

Indeed, a straightforward calculation shows that

$$\|\beta_k\|^2 / \phi_k + \tilde{w}_k \phi_k = \left(\|\beta_k\| / \phi_k^{1/2} - \tilde{w}_k^{1/2} \phi_k^{1/2} \right)^2 + 2\tilde{w}_k^{1/2} \|\beta_k\| \quad (3.68)$$

from which (3.67) follows immediately. Inserting (3.67) in (3.66) yields the following minimization problem in the auxiliary variable vector β :

$$\begin{aligned} & \min_{\beta} \sum_{k=1}^{M+1} \tilde{w}_k^{1/2} \|\beta_k\| \\ & \text{s.t. } A\beta = y \end{aligned} \quad (3.69)$$

which is an SOCP [16].

Solving (3.69) for β and then obtaining $\hat{\phi}$ from (3.67) is a much more efficient means of computing the solution of (3.59) than solving the SDP in (3.61). However, we have observed empirically that the following Cyclic Algorithm (CA) for solving the augmented problem (3.62) is an even faster way to obtain (or approximate, see below) the sought vector $\hat{\phi}$.

CA

a) With ϕ fixed at its most recent estimate $\hat{\phi}$, compute the minimizer of (3.62) with respect to β (see (3.63)) :

$$\hat{\beta} \leftarrow \hat{P}A^*\hat{R}^{-1}y \quad (3.70)$$

where \hat{P} and \hat{R} are made from $\hat{\phi}$.

b) Update $\hat{\phi}$ by computing the minimizer of (3.62) with respect to ϕ , for β fixed at $\hat{\beta}$ (see (3.67)):

$$\hat{\phi}_k \leftarrow \|\hat{\beta}_k\|/\tilde{w}_k^{1/2} \quad k = 1, \dots, M+1 \quad (3.71)$$

c) Iterate the above two steps until a convergence criterion is satisfied.

The CA decreases the function in (3.62) (and hence $h(\phi)$) at each iteration and can be shown to converge globally from any point $\{\phi_k > 0\}$ (see [93] and the references there). However its convergence to the sought solution of (3.62) is linear in general and hence the CA can become rather slow as its iterations approach the solution. Consequently, it can be advisable to use a rough convergence criterion in step c) above, for example to stop the CA after a fixed number of iterations (we use this idea in the numerical example section where we stop the CA after ten iterations). Note that this is an added bonus of using the CA in lieu of an off-the-shelf SOCP solver, as typically the user cannot modify the stopping criterion of the latter solver.

Using $h(\phi)$ to majorize $f(\phi)$ for any given $\tilde{\phi}$, and the CA to decrease $h(\phi)$ from $h(\tilde{\phi})$ to $h(\hat{\phi})$ (and hence $f(\phi)$ from $f(\tilde{\phi})$ to $f(\hat{\phi})$) yields the LIMES algorithm whose steps are summarized below.

LIMES

Initialization. Use DAM to compute an initial spectral estimate $\tilde{\phi}$ (i.e., $\tilde{\phi}_k$ is given by $\phi(\mu_s)$ in (3.15) with $s = \arg \min_p |\mu_p - \omega_k|$).

Main step. Using the most recent value of $\tilde{\phi}$ to compute the weights $\{\tilde{w}_k\}$, employ the CA (initialized with $\tilde{\phi}$) to obtain $\hat{\phi}$.

Iteration/Exit. If the convergence criterion is satisfied then exit ; Else set $\tilde{\phi} \leftarrow \hat{\phi}$ and return to the main step (let $\hat{\phi}^i$ denote the estimate obtained at the i -th iteration of the main step; we have used the following convergence criterion in the numerical examples of the next section : if $\|\hat{\phi}^{i+1} - \hat{\phi}^i\|/\|\hat{\phi}^i\| \leq 10^{-3}$ then we consider that LIMES has converged).

Presumably if the CA is stopped before convergence (as suggested above) then

the number of main steps that need to be executed to ensure the convergence of LIMES will increase. However, it was observed that this increase is not significant, which can be at least partly explained by the fact that the initialization of CA with the most recent estimate of ϕ (as done above) is usually quite good and thus the algorithm needs only a few steps to approach the minimizer of $h(\phi)$. (once again in contrast to this, the user of an off-the-shelf SOCP solver has no control over its initialization). Consequently the overall computational burden of LIMES may well decrease when a rough convergence criterion is used in the CA.

3.7 Numerical illustrations and concluding remarks

A. Band-limited CARMA signals

One says that $y(t)$ is a band-limited Continuous-time Auto-Regressive Moving-Average (CARMA) signal if its spectrum has the following form :

$$\phi(\omega) = \begin{cases} |\gamma(i\omega)|^2 / |\delta(i\omega)|^2 & |\omega| \leq B \\ 0 & \text{else} \end{cases} \quad (3.72)$$

where $\gamma(s)$ and $\delta(s)$ are polynomials in the complex variable s , and B is the bandwidth (in [rad/sec]). We will consider two instances of such CARMA signals (see below) : one with $\delta(s) = 1$ that has a smooth spectrum, and another with $\delta(s) \neq 1$ which comparatively has a less smooth spectrum. Let $\tilde{B} = B/2\pi$ denote the bandwidth in [Hz]. In both cases we let \tilde{B} take on the following three values: $\tilde{B} = 1$; 2.5 ; and 5 [Hz] (note that we will choose the polynomial coefficients such that $|\gamma(i\omega)|^2 / |\delta(i\omega)|^2$ is almost zero at $|\omega| = B$, see below).

The sampling intervals $\{t_{n+1} - t_n\}_{n=1}^{N-1}$ [sec], with $N = 500$, are shown in Figure 3.1: $\{t_n\}$ have been generated as random variables uniformly distributed in $[0, 500]$, which were rounded-off to two significant decimals; hence one can expect that $\min_n(t_{n+1} - t_n)$ is on the order of 10^{-2} . The same sampling scheme, $\{t_n\}$, is used in both CARMA cases mentioned above. The corresponding spectral window, $S(v)$, is displayed in Figure 3.2. From this figure we can see that the scheme is alias-free for frequencies up to 50 [Hz] (which is in agreement with the discussion following (3.8)) and it is therefore appropriate for signals with the bandwidth values indicated above.

Using the expression for $\phi(\omega)$ in (3.72), with given polynomials $\gamma(s)$ and $\delta(s)$, we can compute R (see (3.5)) by means of a numerical integration method ; for example, we can use the formula (3.30) for R with $M \gg 1$ such as $M = 10^3$. Once R is determined we can generate the data vector as :

$$y = R^{1/2} \varepsilon \quad (3.73)$$

where $\varepsilon \in \mathbb{C}^N$ comprises a sequence of uncorrelated normal random variables with zero mean and unit variance. Different realizations of ε yield different data vectors y that are then used in the Monte-Carlo simulation-based study of LIMES and DAM in the following examples.

All-zero example

Let

$$\begin{aligned}\gamma(s) &= (0.5s + i0.5)(0.5s - i0.5)/(\pi)^2 \\ \delta(s) &= 1\end{aligned}\quad (3.74)$$

The bandwidth of the corresponding signal is $\tilde{B} = 1$ [Hz]. To extend the bandwidth to the other values mentioned above, viz. $\tilde{B} = 2.5$ [Hz] and $\tilde{B} = 5$ [Hz] (while maintaining the shape of the spectrum), we scale the coefficients of $\gamma(s)$ in (3.74) to obtain the following CARMAs :

$$\begin{aligned}\gamma(s) &= (0.2s + i0.5)(0.2s - i0.5)/(\pi)^2 \\ \delta(s) &= 1\end{aligned}\quad (3.75)$$

(with $\tilde{B} = 2.5$ [Hz]) and, respectively,

$$\begin{aligned}\gamma(s) &= (0.1s + i0.5)(0.1s - i0.5)/(\pi)^2 \\ \delta(s) &= 1\end{aligned}\quad (3.76)$$

(with $\tilde{B} = 5$ [Hz]). The true spectra of the corresponding band-limited CARMA signals are shown in Figure 3.3. In the same figure we present the means and standard deviations of the spectral estimates obtained in 100 Monte-Carlo runs by using DAM and LIMES with $M = 20$.

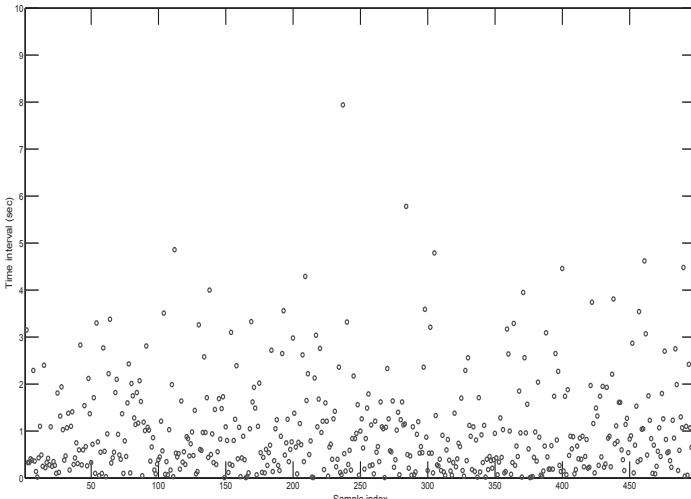


Figure 3.1. The sampling intervals $\{t_{n+1} - t_n\}_{n=1}^{499}$ [sec]

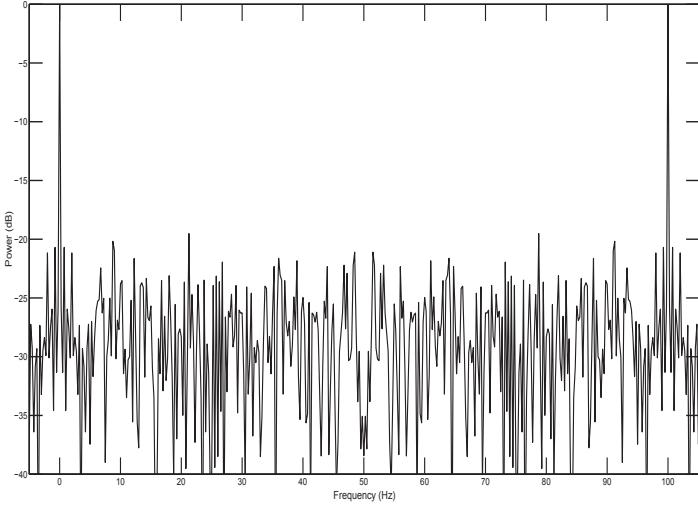
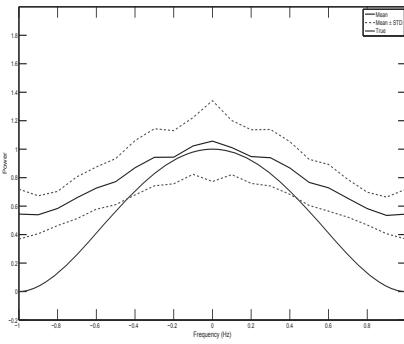


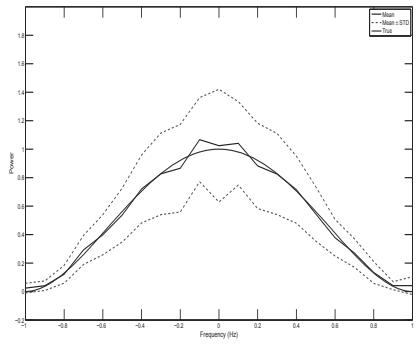
Figure 3.2. The spectral window $S(v)$ [dB] for the $\{t_n\}_{n=1}^{500}$ corresponding to Figure 3.1.

From Figure 3.3, we observe that the DAM estimate is heavily biased at high frequencies in all cases considered. This was expected in view of the leakage from the low-frequency bands that have a larger power. Furthermore, the bias of the DAM estimate is seen to become significant even at low frequencies as B increases, and this kind of behavior begs for a more detailed explanation. Of course, leakage can still be used to explain it because more and more power leaks into any given frequency band as B increases. However it would also be interesting to understand why the DAM estimate is biased in the manner illustrated in Figure 3.3 where this estimate tends to be nearly constant at all frequencies, as B increases, regardless of the shape of the true spectrum. The explanation of this behavior lies in the fact that the samples $\{y(t_n)\}$ of the signal, for given $\{t_n\}$, tend to become less and less correlated as B increases ; consequently, $\{y(t_n)\}$ provide less and less information about the correlation properties of the signal (unless $\{t_n\}$ are decreased proportionately to $1/B$) and thus the signal appears to be white. This observation explains the behavior of the DAM estimate which can determine only the power of the signal, but not the shape of its spectrum, as B increases.

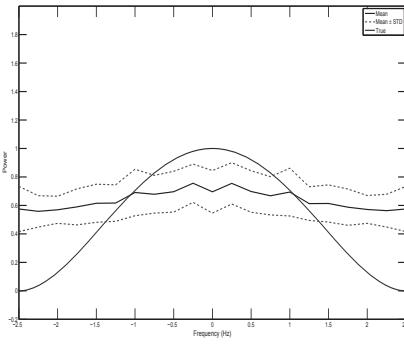
In sharp contrast with the poor performance of DAM, the LIMES estimate degrades quite gracefully as B increases (presumably owing to its ML character). In particular LIMES is nearly unbiased even for the largest value of B considered in Figure 3.3. The root-MSE of LIMES is compared with the root-CRB in Figure 3.4 (DAM is not considered in this comparison because, as explained above, it becomes more or less useless as a spectral estimation method



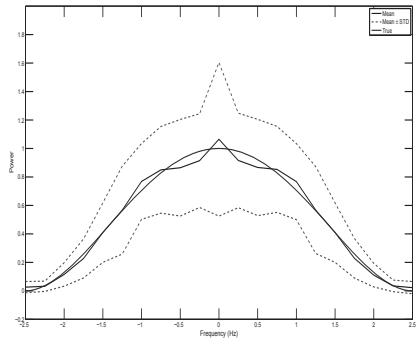
a) DAM ($\tilde{B} = 1$ Hz)



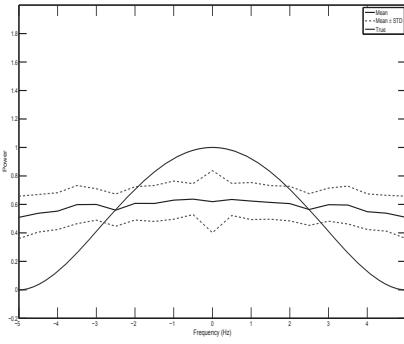
b) LIMES ($\tilde{B} = 1$ Hz)



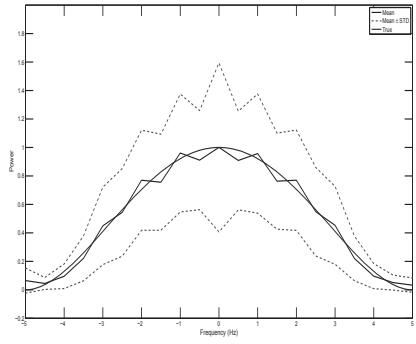
c) DAM ($\tilde{B} = 2.5$ Hz)



d) LIMES ($\tilde{B} = 2.5$ Hz)



e) DAM ($\tilde{B} = 5$ Hz)



f) LIMES ($\tilde{B} = 5$ Hz)

Figure 3.3. The true spectrum (thick solid line) along with the mean (thin solid line) and the mean \pm one standard deviation (dashed lines) for the DAM estimate and the LIMES estimate with $M = 20$.

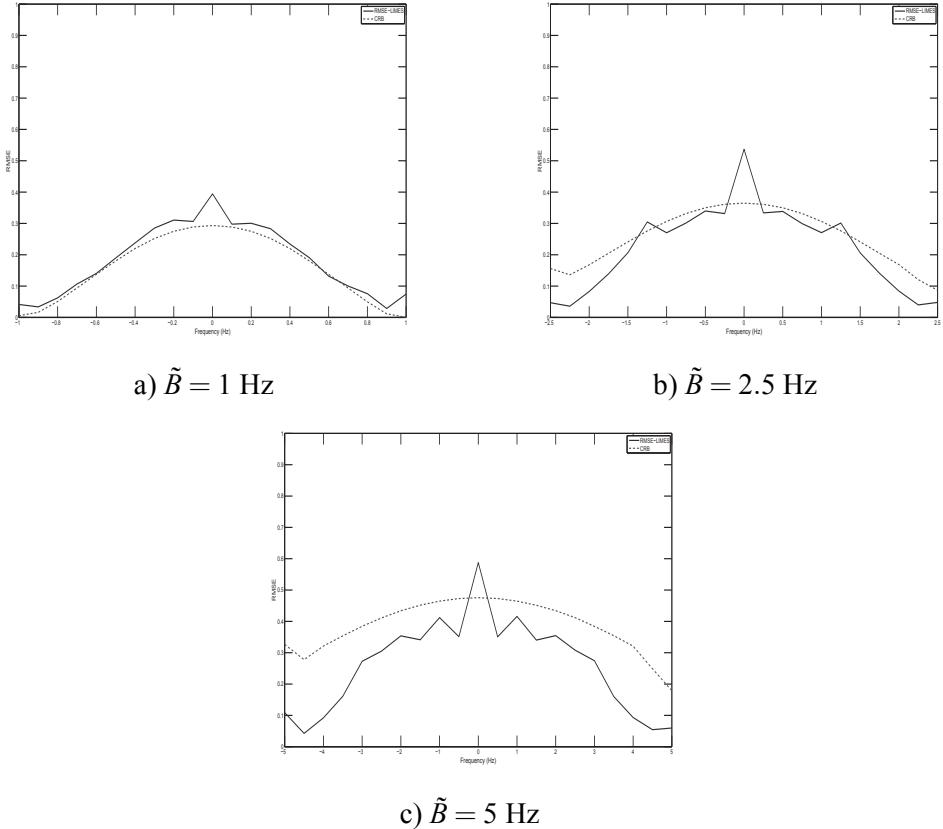


Figure 3.4. The root-MSEs of LIMES estimate (solid lines), with $M = 20$, compared with the root-CRB (dashed lines).

when B increases). We can see from this figure that the MSE of LIMES follows closely the CRB for $\tilde{B} = 1$ and $\tilde{B} = 2.5$ [Hz], whereas it tends to be smaller than the CRB for $B = 5$ [Hz] presumably due to a (small) bias.

Regarding the computational complexities of the two methods under consideration, as already mentioned above DAM is (much) faster than LIMES. More concretely, the average time per realization in the above example was 1 sec for DAM and 9 min for LIMES.

Mixed zero-pole example

In this sub-section we consider the following three CARMAs whose spectra have the same shape but each has a different bandwidth as also assumed in the previous example:

$$\begin{aligned}\gamma(s) &= 0.25s^2 + 2.5s + 2.792 \\ \delta(s) &= (0.5s + 0.2)(0.5s + 0.25 + i\pi/2)(0.5s + 0.25 - i\pi/2)\end{aligned}\quad (3.77)$$

(with $\tilde{B} = 1$ [Hz]),

$$\begin{aligned}\gamma(s) &= 0.04s^2 + s + 2.792 \\ \delta(s) &= (0.2s + 0.2)(0.2s + 0.25 + i\pi/2)(0.2s + 0.25 - i\pi/2)\end{aligned}\quad (3.78)$$

(with $\tilde{B} = 2.5$ [Hz]), and

$$\begin{aligned}\gamma(s) &= 0.01s^2 + 0.5s + 2.792 \\ \delta(s) &= (0.1s + 0.2)(0.1s + 0.25 + i\pi/2)(0.1s + 0.25 - i\pi/2)\end{aligned}\quad (3.79)$$

(with $\tilde{B} = 5$ [Hz]).

The corresponding true spectra along with the means and means \pm standard deviations of their estimates obtained by DAM and LIMES with $M = 40$ are shown in Figure 3.5. Furthermore, Figure 3.6 compares the root-MSE of LIMES with the root-CRB. From these figures we can draw the same observations about the performance of DAM and LIMES as in the previous example, in spite of the fact that the CARMA signals considered here have a much larger dynamic range (i.e. they are less smooth).

Based on our experience with the above and other similar numerical examples, it appears that LIMES can provide satisfactory performance for a bandwidth B up to half the folding frequency determined from the spectral window, whereas DAM typically requires a much smaller B (e.g. twenty times smaller than the folding frequency).

Regarding the choice of M , this can also be done solely from data by means of the Akaike information criterion (AIC) which selects M as the solution of the following minimization problem :

$$\min_M \ln |\hat{R}_M| + y^* \hat{R}_M^{-1} y + M \quad (3.80)$$

where \hat{R}_M is the matrix R made from the LIMES estimates of $\{\phi_k\}_{k=1}^{M+1}$. The use of AIC yields :

$$M = 20 \text{ for (3.74) (with } \tilde{B} = 1 \text{ Hz)} \quad (3.81)$$

and

$$M = 40 \text{ for (3.77) (with } \tilde{B} = 1 \text{ Hz)} \quad (3.82)$$

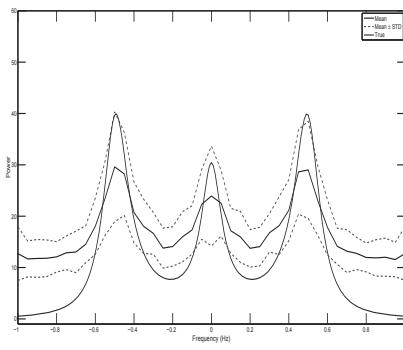
which are the values of M used in the previous examples. For the larger values of \tilde{B} considered in the said examples, AIC recommends smaller values for M yet we chose to use the values in (3.81) and, respectively, (3.82) for simplicity (note that, as \tilde{B} increases, the variance of LIMES appears to increase faster than the bias, see Figures 3.3 and 3.5, and therefore one can expect that M should indeed be decreased to trade-off bias for variance).

Remark 2 *Besides the above examples, we have performed several other numerical experiments the details of which are omitted for conciseness; however we would like to summarize here some of the conclusions drawn from these additional examples: i) LIMES works well even in the cases where B is over-estimated; ii) the proposed method is quite insensitive to relatively large side-lobes of the spectral window (the so-called "pseudo-aliasing" window case), unlike DAM which suffers a considerable accuracy loss in such a situation; and iii) LIMES performs reasonably well even for "non-smooth" spectra, provided that these spectra can be well described by the piecewise linear model in (3.10) with a relatively small value of M (i.e. $M \ll N$).*

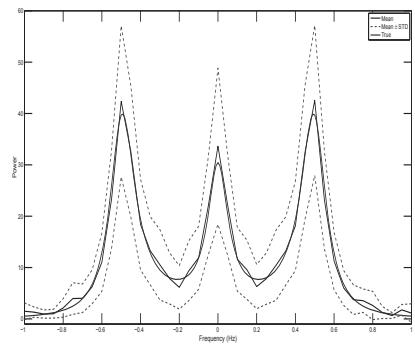
B. Ocean wave data

The data set considered in this sub-section consists of measurements of the sea level, at a location in the Pacific Ocean, obtained using a wave follower [76] [48]. Because the sensor cannot reliably measure frequencies higher than 1 Hz, the recorded analog signal was pre-processed using a low-pass filter with a cut-off frequency of about 1 Hz; the filtered signal was then uniformly sampled at a rate of 4 Hz to obtain 1024 data points. The main goal of spectrally analyzing this data set is to investigate whether the spectrum decreases from 0.2 Hz to 1 Hz at a rate which is consistent with that predicted by a physical wave model. The frequency range $[0, 0.2]$ [Hz] is not of direct interest to this analysis. On the other hand, the interval $[1, 2]$ [Hz], while of little relevance to the physical wave study, is of interest to verify the frequency characteristics of the sensor and of the low-pass filter used [76] [48]. With this fact in mind we set $\tilde{B} = 2$ [Hz]; given the uniform sampling rate of 4 Hz, this is in fact the maximum possible alias-free frequency.

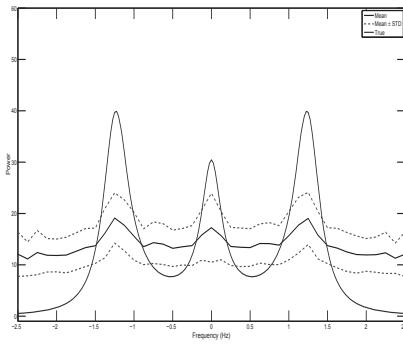
The spectra estimated from all $N = 1024$ data samples via DAM and LIMES with $M = 50$ are shown in Figure 3.7. In this figure, the LIMES spectral estimate appears to be affected by much less leakage than the DAM estimate; the leakage-induced bias of the latter is particularly evident in the high-frequency



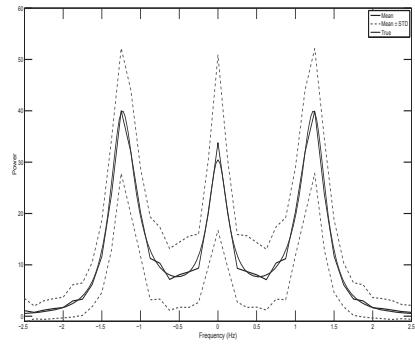
a) DAM ($\tilde{B} = 1$ Hz)



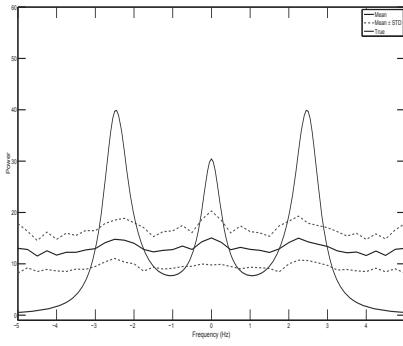
b) LIMES ($\tilde{B} = 1$ Hz)



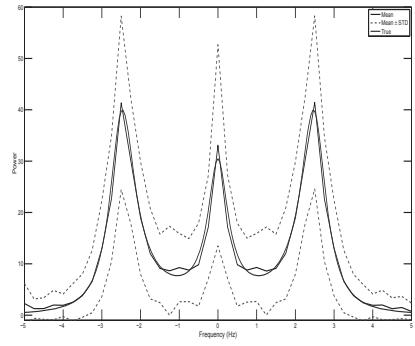
c) DAM ($\tilde{B} = 2.5$ Hz)



d) LIMES ($\tilde{B} = 2.5$ Hz)



e) DAM ($\tilde{B} = 5$ Hz)



f) LIMES ($\tilde{B} = 5$ Hz)

Figure 3.5. The true spectrum (thick solid line) along with the mean (thin solid line) and the mean \pm one standard deviation (dashed lines) for the DAM estimate and the LIMES estimate with $M = 40$.

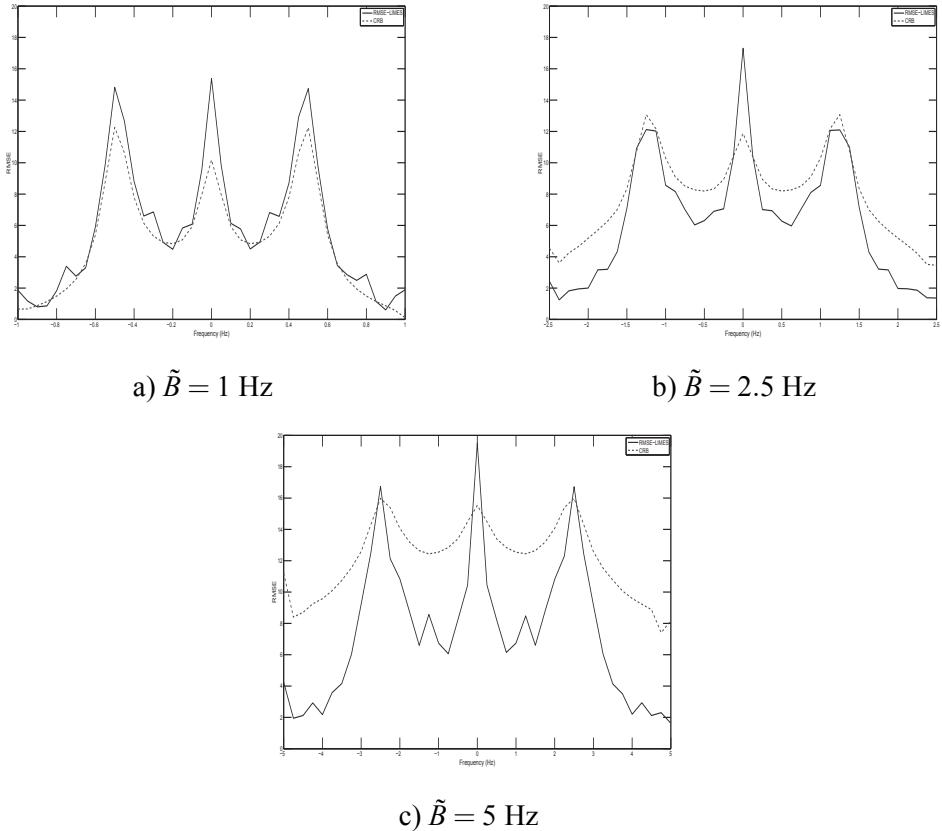


Figure 3.6. The root-MSEs of LIMES estimate (solid lines), with $M = 40$, compared with the root-CRB (dashed lines).

range $[1, 2]$ [Hz]. The LIMES estimate is also smoother than the DAM estimate in the range of main interest $[0.2, 1]$ [Hz], which makes slope determination from the LIMES plot a more straightforward task.

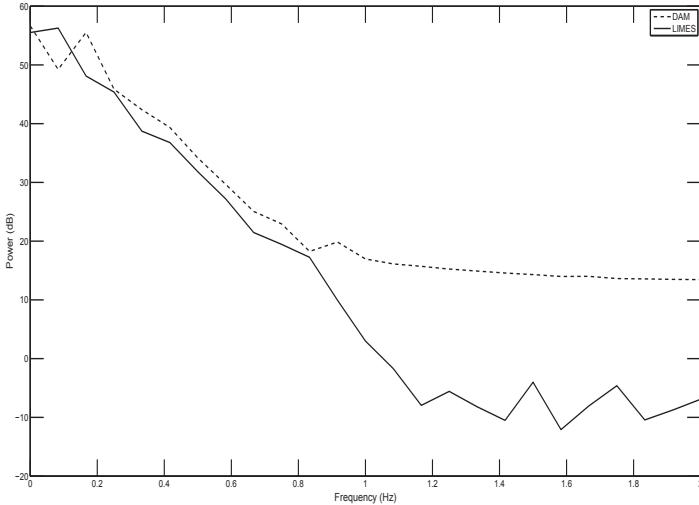


Figure 3.7. The DAM and LIMES spectral estimates for $N = 1024$ and $M = 50$ (uniform data case, one realization).

Because the main topic of this chapter is spectral analysis from irregularly-sampled data, we have also applied the DAM and LIMES methods (with $M = 20$) to ten data sets consisting of $N = 400$ samples randomly selected from the total of 1024 available data points (note that such a random decimation (or down-sampling) operation yields a so-called missing data set, which is a special case of irregularly sampled signal). Figure 3.8 shows the average of the DAM and LIMES spectral estimates, obtained from these ten data sets, along with their standard deviations. Despite the much smaller number of data samples processed, the LIMES estimate is quite accurate (compare it with that in Figure 3.7) and easy to use for slope determination in the interval $[0.2, 1]$ [Hz], whereas the DAM estimate is useless owing to a significant leakage-induced bias. To explain the leakage problem of the DAM spectral estimate (see also the CARMA examples for further details on this aspect), we show in Figure 3.9 the spectral windows corresponding to the uniform sampling instance of Figure 3.7 and to one of the ten cases of non-uniform sampling that are behind Figure 3.8. As can be seen from Figure 3.9, the sidelobes of the spectral window are much larger in the nonuniform data case than in the uniform one, which explains the serious leakage problem of the DAM estimate in the nonuniform case; in contrast with this LIMES, which estimates all spectral components jointly, has no leakage problems.

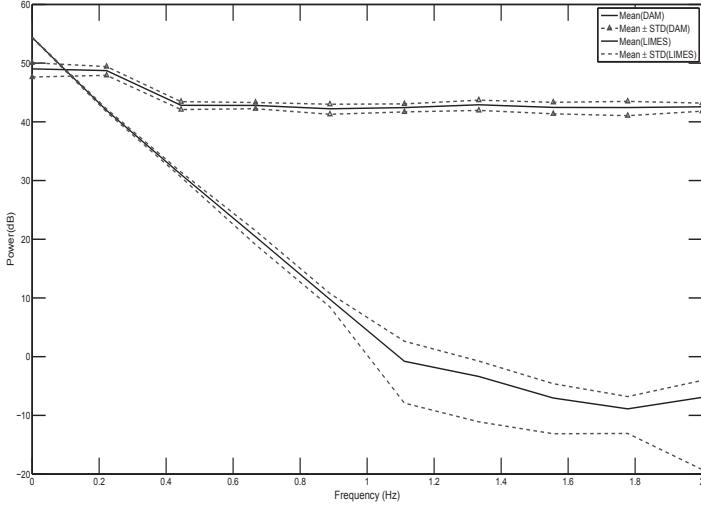


Figure 3.8. Mean (solid line) and mean \pm one standard deviation (dashed lines) spectral plots for the DAM and LIMES estimates with $N = 400$ and $M = 20$ (nonuniform data case, ten realizations).

3.8 Evaluation of the integrals in (3.18) and (3.19)

For $\tau = 0$:

$$\begin{aligned} F_{ps}(k+1) &= \frac{1}{2\Delta} (\omega - \omega_k)^2 \Big|_{\omega_k}^{\omega_{k+1}} = \Delta/2 \quad (p = s) \\ G_{ps}(k) &= -\frac{1}{2\Delta} (\omega - \omega_{k+1})^2 \Big|_{\omega_k}^{\omega_{k+1}} = \Delta/2 \quad (p = s) \end{aligned} \quad (3.83)$$

For $\tau \neq 0$ we use the standard formula for integration by parts, viz.

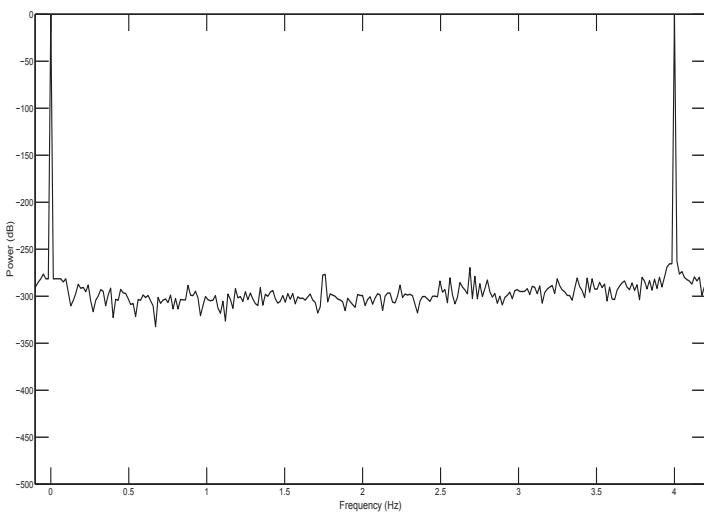
$$\int_{\omega_k}^{\omega_{k+1}} u(\omega) v'(\omega) d\omega = u(\omega) v(\omega) \Big|_{\omega_k}^{\omega_{k+1}} - \int_{\omega_k}^{\omega_{k+1}} u'(\omega) v(\omega) d\omega \quad (3.84)$$

with $u(\omega) = \omega - \omega_k$ for $F_{ps}(k+1)$ and $u(\omega) = \omega - \omega_{k+1}$ for $G_{ps}(k)$, and with $v'(\omega) = dv(\omega)/d\omega = e^{i\omega\tau}$, to obtain :

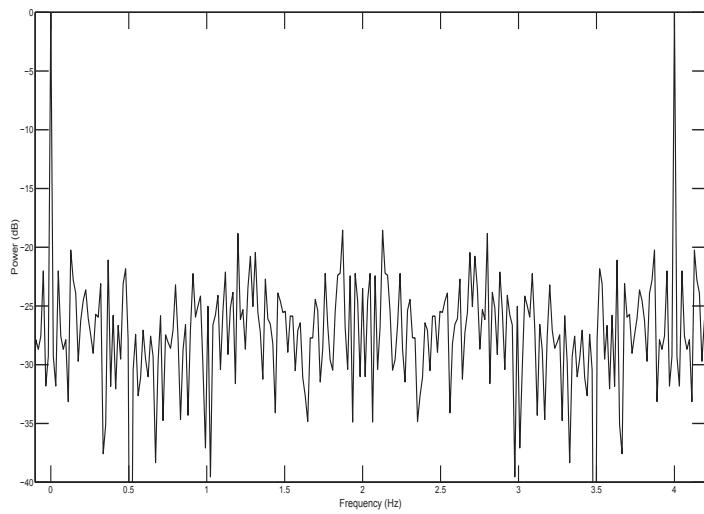
$$\begin{aligned} F_{ps}(k+1) &= \frac{\omega - \omega_k}{\Delta} \frac{e^{i\omega\tau}}{i\tau} \Big|_{\omega_k}^{\omega_{k+1}} - \frac{1}{i\tau\Delta} \int_{\omega_k}^{\omega_{k+1}} e^{i\omega\tau} d\omega \\ &= \frac{e^{i\omega_{k+1}\tau}}{i\tau} + \frac{1}{\tau^2\Delta} (e^{i\omega_{k+1}\tau} - e^{i\omega_k\tau}) \\ &= e^{i\omega_{k+1}\tau} \left[\frac{1}{i\tau} + \frac{1}{\tau^2\Delta} (1 - e^{-i\Delta\tau}) \right] \quad (p \neq s) \end{aligned} \quad (3.85)$$

and

$$\begin{aligned} G_{ps}(k) &= -\frac{\omega - \omega_{k+1}}{\Delta} \frac{e^{i\omega\tau}}{i\tau} \Big|_{\omega_k}^{\omega_{k+1}} + \frac{1}{i\tau\Delta} \int_{\omega_k}^{\omega_{k+1}} e^{i\omega\tau} d\omega \\ &= -\frac{e^{i\omega_k\tau}}{i\tau} - \frac{1}{\tau^2\Delta} (e^{i\omega_{k+1}\tau} - e^{i\omega_k\tau}) \\ &= e^{i\omega_k\tau} \left[-\frac{1}{i\tau} - \frac{1}{\tau^2\Delta} (e^{i\Delta\tau} - 1) \right] \quad (p \neq s) \end{aligned} \quad (3.86)$$



a) Uniform sampling



b) Nonuniform sampling

Figure 3.9. The spectral window $S(v)$ for the uniform sampling case and for one instance of nonuniform sampling.

3.9 Algebraic proof of (3.57)

Letting

$$Z = R^{1/2} \tilde{R}^{-1} R^{1/2} \quad (3.87)$$

where $R^{1/2}$ is a Hermitian square-root of R , we can re-write (3.57) as:

$$\ln|Z| + N \leq \text{tr}(Z) \quad (3.88)$$

or equivalently

$$\sum_{k=1}^N (\lambda_k - \ln\lambda_k - 1) \geq 0 \quad (3.89)$$

with $\{\lambda_k > 0\}$ being the eigenvalues of the positive definite matrix Z . The generic term of the sum in (3.89) has the form:

$$u(\lambda) = \lambda - \ln\lambda - 1, \lambda > 0 \quad (3.90)$$

The first and second derivatives of the above function are given by

$$\begin{aligned} u'(\lambda) &= 1 - 1/\lambda \\ u''(\lambda) &= 1/\lambda^2 \end{aligned} \quad (3.91)$$

Therefore $u(\lambda)$ has a unique minimum at $\lambda = 1$ whose value is $u(1) = 0$. This means that $u(\lambda) \geq 0, \forall \lambda > 0$, which proves (3.89) and thus the inequality (3.57).

3.10 Proof of (3.63) and (3.64)

It can be readily verified that

$$\hat{\beta}^* P^{-1} \hat{\beta} = y^* R^{-1} A P P^{-1} P A^* R^{-1} y = y^* R^{-1} y \quad (3.92)$$

which proves (3.64). It remains to show that

$$\beta^* P^{-1} \beta \geq y^* R^{-1} y \quad \text{s.t. } A\beta = y \quad (3.93)$$

Making use of the constraint on β , we can re-write the above inequality as follows:

$$\beta^* (P^{-1} - A^* R^{-1} A) \beta \geq 0 \quad (3.94)$$

which holds if we can prove that the matrix $P^{-1} - A^* R^{-1} A$ in (3.94) is positive semi-definite, i.e. notationally

$$P^{-1} - A^* R^{-1} A \geq 0 \quad (3.95)$$

By a standard result on partitioned positive semi-definite matrices (see, e.g., [52]), (3.95) is equivalent to:

$$\begin{bmatrix} P^{-1} & A^* \\ A & R \end{bmatrix} \geq 0 \quad (3.96)$$

Using the fact that $R = APA^*$ we can re-write (3.93) in the following form

$$\begin{bmatrix} P^{-1/2} \\ AP^{1/2} \end{bmatrix} \begin{bmatrix} P^{-1/2} P^{1/2} A^* \\ \end{bmatrix} \geq 0 \quad (3.97)$$

Because (3.97) is evidently true, the proof is concluded.

4. A nonparametric approach to estimation of smooth spectra from nonuniformly sampled data - enhanced version

4.1 Introduction and brief review of LIMES

Let $y(t) \in \mathbb{R}$ (for $t \in \mathbb{R}$) be a zero-mean stationary continuous-time signal whose power spectral density (or spectrum, for short) is denoted by $\phi(\omega)$ ($\omega \in (-\infty, \infty)$ being the frequency variable). Let $\{y(t_k)\}_{k=1}^N$ be the available data samples. Assume that the spectrum is band-limited to a given frequency $\omega = B$ and that the sampling pattern $\{t_k\}$ is such that it allows the alias-free spectral analysis of $y(t)$ up to a frequency that is not smaller than B (this is a weak assumption on $\{t_k\}$ that are otherwise unconstrained ; see the previous chapter for details on this aspect). Finally, for a given $M \geq 1$ (an integer user parameter), let

$$\omega_k = -B + (k-1)\Delta, k = 1, \dots, M+1; \Delta = 2B/M \quad (4.1)$$

and assume that $\phi(\omega)$ varies linearly (or nearly so) for $\omega \in [\omega_k, \omega_{k+1}]$ ($k = 1, \dots, M$). The above assumptions are quite practical; they only require that we know the bandwidth of the signal (or at least an upper bound on it) and that we possess information on the dynamic range of the spectrum that allows us to choose a suitable value of M . Under these assumptions the problem of interest is the estimation of $\{\phi(\omega_p)\}_{p=1}^{M+1}$ from $\{y(t_k)\}_{k=1}^N$.

LIMES estimates $\{\phi(\omega_p)\}$ by minimizing the (normal) negative log-likelihood function of the data vector $y = [y(t_1), \dots, y(t_N)]^T$, that is :

$$\ln |R| + y^T R^{-1} y \quad (4.2)$$

where R is the covariance matrix of y , and $|R|$ is the determinant of R . It can be shown that R is a linear function of $\{\phi(\omega_k)\}$:

$$R = \sum_{k=1}^{M+1} \phi(\omega_k) C_k \quad (4.3)$$

for some appropriately defined matrices $\{C_k\}$ that depend only on $\{\omega_k\}$ and $\{t_k\}$ [90].

The LIMES algorithm is iterative. The bulk of each iteration consists of computing and inverting the $N \times N$ covariance matrix R (evaluated at the most

recent estimates of $\{\phi(\omega_k)\}$. For large values of N this task becomes computationally prohibitive. According to our experience running LIMES in a standard PC requires about 10 min for a value of N equal to :

$$\tilde{N} = 10^3. \quad (4.4)$$

In some applications of spectral analysis, for example in magnetic resonance spectroscopy and in astronomy, N can be much larger than \tilde{N} (e.g., $N \sim 10^4$ or even $N \sim 10^5$). How can we use LIMES in such cases? The next section provides a simple practical answer to this question.

Numerical experience with LIMES has also revealed that the estimates of $\{\phi(\omega_k)\}$, while virtually unbiased, have a relatively large variance for values of $N = 500$ (which are common in many applications), a fact that can be intuitively explained by the non-parametric character of LIMES (and hence the need for estimating many unknowns from a limited number of data samples). How can we decrease LIMES variance in these cases without compromising the bias? Section 4.3 presents a solution to this problem. In addition, both Sections 4.2 and 4.3 contain an example that numerically illustrates the performance of the proposed solutions.

4.2 Averaged LIMES for very large data sets

Following the main idea of Welch approach to the spectral analysis of regularly sampled data (e.g. [115] [98]), we split the data string $\{y(t_k)\}_{k=1}^N$ in segments of length \tilde{N} with about 50% overlapping between adjacent segments. For example for $\tilde{N} = 10^3$ and $N = 10^4$ we can consider an overlapping factor of 55% (to utilize all data samples). In which case the s -th segment will comprise the following observations :

$$\{y(t_k)\} \text{ for } k = 450(s-1) + p, p = 1, \dots, 10^3; s = 1, \dots, 21 \quad (4.5)$$

We then apply LIMES to each segment, carefully monitoring the condition number of R , $\text{cond}(R)$, during the iterations. The need for monitoring $\text{cond}(R)$ arises from the fact that for some segments with unfavorable sampling patterns $1/\text{cond}(R)$ can be rather close to the machine epsilon, in which case the computations in the LIMES algorithm can be affected by considerable numerical errors. As a consequence of this observation, we *discard* the results associated with any data segment for which $\text{cond}(R) > 10^{12}$. Then we compute the average (and the standard deviation, if desired) of the LIMES estimates obtained from the remaining segments. The averaged spectral estimate, computed as outlined above, is used as the estimated spectrum in lieu of the LIMES estimate based on all N data samples (whose computation was deemed to be intractable).

Numerical example

We consider the spectrum :

$$\phi(\omega) = |C(i\omega)|^2 \quad (4.6)$$

where $C(s)$ is the following second-order polynomial

$$C(s) = (0.5s + i0.5)(0.5s - i0.5)/\pi^2 \quad (4.7)$$

The bandwidth of this spectrum is $B/2\pi = 1$ [Hz]. Furthermore, this spectrum has a relatively small dynamic range and thus we can use a reasonably small value for M , viz. $M = 20$. Samples $\{y(t_k)\}$ of a continuous-time moving average signal with the above type of spectrum can be generated as described in [90]. Let $N = 10^4$ and let $\{t_k\}$ be random variables that are uniformly distributed in the interval $[0, 10^4]$. Computing the LIMES estimate from all $N = 10^4$ samples using a standard PC is an intractable task. Consequently, we split the data string in 21 segments with 55% overlapping, as in (4.5), and applied LIMES to each of those segments. Six of the segments did not pass the test on $\text{cond}(R)$. The *averaged LIMES estimate* (obtained from the retained 15 segments) and its standard deviation are shown (for a typical data realization) in Figure 4.1, along with the true spectrum. We can see from the figure that the estimate is quite close to the true spectrum.

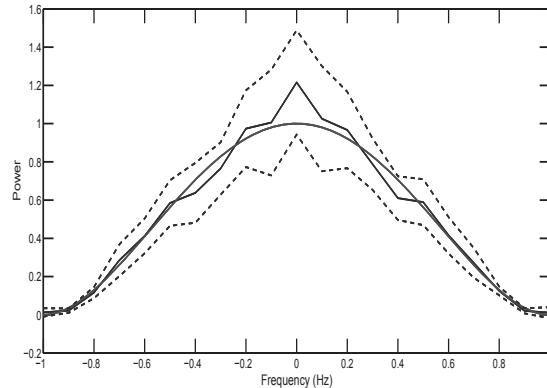


Figure 4.1. The true spectrum (thick solid line) and the averaged LIMES estimate (thin line) ± 1 standard deviation curves (dotted lines).

4.3 Smoothed LIMES for small/medium data sets

As mentioned in Section 4.1, for values of $N \leq 500$ and for $M \geq 20$, the LIMES estimates can have a relatively large variances, presumably owing to

the need for estimating a large number of unknowns from a limited number of data samples. When the true spectrum has a relatively small dynamic range the variance of LIMES estimation errors can be reduced via smoothing without compromising the bias significantly. We propose using a polynomial (i.e. moving average) spectrum for the purpose, which we will fit to the LIMES spectral estimates $\{\hat{\phi}(\omega_k)\}_{k=1}^{M+1}$ in a least-squares (LS) sense. More concretely, we will re-estimate the spectrum as $|\hat{C}(i\omega)|^2$ where $\hat{C}(s)$ is an n -order polynomial, $\hat{C}(s) = \hat{c}_0 + \hat{c}_1 s + \cdots + \hat{c}_n s^n$, that is obtained from the minimization of the following LS fitting criterion:

$$\sum_{k=1}^{M+1} [\hat{\phi}(\omega_k) - |C(i\omega_k)|^2]^2 \quad (4.8)$$

To solve this minimization problem, we note that $|C(i\omega)|^2 = C(s)C(-s)|_{s=i\omega}$ and that $C(s)C(-s)$ is a $2n$ -order polynomial (below we assume that $c_k = 0$ if $k \notin [0, n]$):

$$\begin{aligned} C(s)C(-s) &= \sum_{p=0}^n \sum_{k=0}^n c_k c_p s^k s^p (-1)^p = \sum_{p=0}^n \sum_{m=p}^{p+n} c_p c_{m-p} s^m (-1)^p \\ &= \sum_{m=0}^{2n} a_m s^m; a_m = \sum_{p=0}^n c_p c_{m-p} (-1)^p. \end{aligned} \quad (4.9)$$

Because $C(s)C(-s)$ is real-valued for $s = i\omega$, it must be true that

$$a_m = 0 \text{ for } m = 1, 3, 5, \dots, 2n-1 \quad (4.10)$$

This property can in effect be directly shown as follows : if m is odd then

$$a_m = \sum_{k=m-n}^m c_k c_{m-k} (-1)^m (-1)^k = - \sum_{k=0}^n c_k c_{m-k} (-1)^k = -a_m \quad (4.11)$$

which proves (4.10). Let

$$\psi_n(\omega) = [1 \ \omega^2 \ \omega^4 \ \cdots \ \omega^{2n}]^T \quad (n+1) \times 1 \quad (4.12)$$

$$\theta_n = [a_0 \ a_2 \ a_4 \ \cdots \ a_{2n}]^T \quad (n+1) \times 1 \quad (4.13)$$

Using this notation, along with the assumptions made above, see (4.9) and (4.10), we can re-write the LS criterion in (4.8) as a quadratic function of θ_n :

$$V(\theta_n) = \sum_{k=1}^{M+1} [\hat{\phi}(\omega_k) - \psi_n^T(\omega_k) \theta_n]^2. \quad (4.14)$$

The minimizer, $\hat{\theta}_n$, of (4.14) can be readily computed by solving a (over-determined) system of linear equations (see, e.g. [98]). Furthermore, a suitable

value for the order n can be determined by minimizing the following Akaike information criterion (AIC) (see, e.g. [3] [102]):

$$\text{AIC}_n = \ln[V(\hat{\theta}_n)] + \frac{2n}{M+1} \quad (4.15)$$

Remark 3 *The polynomial in (4.9) is not only real valued for $s = i\omega$, but also non-negative. This fact imposes further constraints on $\{a_m\}$ besides the one in (4.10). These constraints can be satisfied by parameterizing the coefficients $\{a_m\}$ in a special way, as a functions of a positive semi-definite matrix (see, e.g., [73]). If this is done then (4.14) is no longer a simple linear LS problem but a semi-definite program (SDP). While such an SDP can still be solved globally, we have observed that it is rather ill-conditioned numerically even for relatively small values of n . For this reason we do not pursue the SDP idea here, and therefore we will constrain $\{a_m\}$ only as in (4.10). While this means that $\sum a_m(i\omega)^m$ can in principle take on negative values for some ω 's, this is unlikely to happen as the estimated spectrum $\{\hat{\phi}(\omega_k)\}$ in (4.14) to which we fit is always positive.*

Numerical example

We consider the same true spectrum as in (4.6), but this time we let $N = 500$ and accordingly let $\{t_k\}$ be random variables that are uniformly distributed in $[0, 500]$. The spectrum estimated by LIMES in a typical data realization has a relatively large variance, see Figure 4.2. Figure 4.3 shows the plot of AIC_n as a function of n . As can be seen from this figure the values of n recommended by AIC is equal to two (which also happens to be the true order of the moving-average spectrum in (4.6)). The *smoothed LIMES estimate* obtained by minimizing (4.14), viz. $\{\psi_2^T(\omega_k)\hat{\theta}_2\}_{k=1}^{M+1}$, is shown in Figure 4.2 : this estimate is much closer to the true spectrum than the initial LIMES estimate is.

4.4 Concluding remarks

For large values of N (such as $N \sim 10^4$ or larger), the LIMES estimate is quite accurate but it can be expensive to compute. In such cases the averaged LIMES estimate is computationally much more convenient to use without compromising the statistical performance significantly. For smaller values of N (such as $N \sim 10^2$), the variance of the LIMES spectral estimate can be rather large. The variance can be reduced, without trading-off too much bias, by using the smoothed LIMES which should be the method of choice whenever the spectrum is known to have a reasonably small dynamic range.

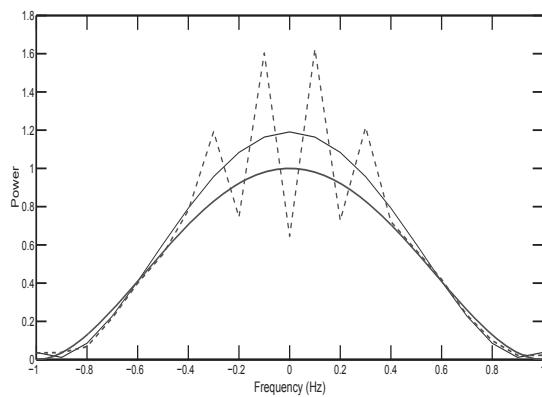


Figure 4.2. The true spectrum (thick line), the initial LIMES estimate (dotted line) and the smoothed LIMES estimate (thin line).

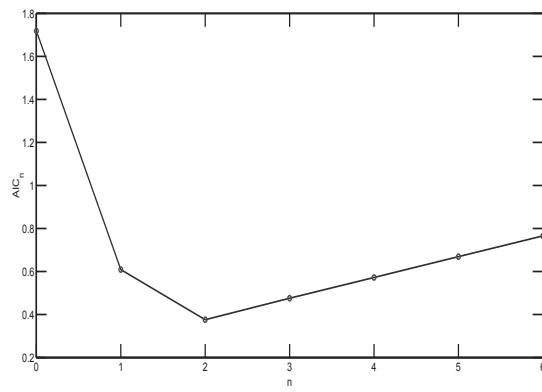


Figure 4.3. AIC_n for $n \in [0, 6]$.

Part II: Sparse methods

5. Spectral analysis of nonuniformly sampled data via a sparse parameter estimation approach: SPICE

5.1 Introduction and problem formulation

Let $y \in \mathbb{C}^{N \times 1}$ denote the available data vector (or snapshot), and consider the following model for y :

$$y = \sum_{c=1}^C a(\tilde{\omega}_c) \tilde{s}_c + \varepsilon \quad (5.1)$$

where $\varepsilon \in \mathbb{C}^{N \times 1}$ is a noise term, $\tilde{s}_c \in \mathbb{C}$ and $\tilde{\omega}_c \in \Omega \subset \mathbb{R}$ are the unknown parameters of the c -th signal component, $a(\cdot) : \Omega \rightarrow \mathbb{C}^{N \times 1}$ is a known function, and C is the unknown number of components. This type of model is a frequent occurrence in numerous applications such as spectral analysis [98] [51], array processing [99], radar imaging [40] [56], astronomy [86] and elsewhere [37], in all of which estimation of the unknown parameters in (5.1) is a basic goal. Note that in some of these applications, for instance in array processing, the number of available snapshots is larger than one. However, to keep the notation and explanations as simple as possible, we will first consider the single snapshot case, but afterwards will also discuss briefly the extension to the multi-snapshot (also called replicated-measurement) case.

The estimation methods associated with (5.1) can be categorized in three large classes: parametric, non-parametric and semi-parametric, see below for details.

A prominent member of the *parametric class* is the nonlinear least squares (NLS) method that consists of minimizing the following criterion :

$$\|y - \sum_{c=1}^C a(\tilde{\omega}_c) \tilde{s}_c\|^2 \quad (5.2)$$

with respect to $\{\tilde{\omega}_c, \tilde{s}_c\}$ (for given C). Observe that the criterion above depends quadratically on $\{\tilde{s}_c\}$, which means that it can be minimized explicitly with respect to these parameters (for fixed $\{\tilde{\omega}_c\}$). In other words, $\{\tilde{s}_c\}$ can be separated out - and this along with the fact that the signal components enter (5.1) through separated terms give the name of *separable models* to (5.1) [37] [36]. The NLS enjoys excellent statistical properties; in particular, under the normal white-noise assumption the minimization of (5.2) produces

the maximum-likelihood estimate that is asymptotically statistically efficient. However, this is true only if C used in (5.2) is the “true” number of components and if (5.2) can be globally minimized ; and both these conditions are difficult to meet in practice (especially the global minimization of (5.2) is a hard task).

At the other end of the method spectrum we find the *non-parametric class*. The most basic method of this class is the single-frequency least-squares (SFLS) method (which is also known under other names, such as the periodogram or beamforming method, depending on the application). To explain this method in general terms, let $\{\omega_k\}_{k=1}^K$ denote a fine grid that covers Ω , and assume that $\{\tilde{\omega}_c\}$ lie on (practically, close to) the grid. This means that there exist k_1, \dots, k_C such that $\tilde{\omega}_c = \omega_{k_c}$ ($c = 1, \dots, C$). Let

$$a_k = a(\omega_k) \quad k = 1, \dots, K \quad (5.3)$$

and also let

$$s_k = \begin{cases} \tilde{s}_c & k = k_c \quad (c = 1, \dots, C) \\ 0 & \text{elsewhere} \end{cases} \quad (5.4)$$

Using this notation we can re-write (5.1) as :

$$y = \sum_{k=1}^K a_k s_k + \varepsilon \quad (5.5)$$

where typically $K \gg N$. The SFLS method estimates s_k , in a one-by-one manner, simply ignoring the presence of the other possible signal components in (5.5) :

$$\hat{s}_k = \frac{a_k^* y}{\|a_k\|^2}. \quad (5.6)$$

where the superscript * denotes the conjugate transpose and $\|\cdot\|$ denotes the Euclidean (ℓ_2) norm. Evidently (5.6) does not have the problems (indicated above) that affect the parametric method of NLS. However, the price for the elimination of these problems is poor statistical accuracy : SFLS suffers from local and global leakage problems ; the local leakage reduces the resolution by making it hard to distinguish between signal components with closely-spaced values of $\{\tilde{\omega}_c\}$, whereas global leakage leads to false-alarm, that is to large values of $\{|\hat{s}_k|\}$ for non-existent components. Note that most (if not all) estimated values $\{\hat{s}_k\}$ in (5.6) will usually be different from zero, which motivates the alternative name of *dense* sometimes used to designate the non-parametric methods. In fact in some applications (e.g., radar imaging), in which the number of expected components in (5.5) is rather large, a dense estimate of $\{s_k\}$ can be preferable to a sparse estimate such as one provided by a parametric model. Consequently, there has been a significant interest in devising non-parametric (or dense) estimation methods for (5.1) that possess superior performance to the SFLS method. In Section 5.2 we will describe briefly one

such enhanced method, called IAA (the **i**terative **a**daptive **a**pproach) [122] [96], which eliminates almost completely the leakage problems of the SFLS method in a fully data-adaptive manner (i.e., without requiring the selection of user parameters).

An intermediate category of methods is the *semi-parametric class*. The sparse estimation methods form an important sub-group of this class. These methods use the non-parametric data model in (5.5) but, reminiscent of the parametric approach, they seek to exploit the information that the vector

$$s = [s_1, \dots, s_K]^T \quad (5.7)$$

in (5.5) is sparse (i.e., it has only a few non-zero elements). An archetypical sparse method consists of estimating $\{s_k\}$ by solving the following ℓ_1 -norm constrained LS problem ([107] [61]) :

$$\min_{\{s_k\}} \left\| y - \sum_{k=1}^K a_k s_k \right\|^2 \quad \text{s.t. } \|s\|_1 \stackrel{\Delta}{=} \sum_{k=1}^K |s_k| \leq \eta \quad (5.8)$$

where $\|\cdot\|_1$ stands for the ℓ_1 norm, and η is a threshold that must be chosen by the user. The ℓ_1 -norm constraint in (5.8) is what induces the sparsity of the solution to (5.8) ([107]), which is a potentially useful feature of this type of estimation methods, provided that their user parameters (e.g. η in (5.8)) are well selected. However this selection is by no means a simple task (clearly it is related to the task of estimating C in the parametric model (5.1)), and quite typically the rules proposed for performing it depend on quantities that are unavailable in applications (such as the noise power, or the "true" value of C) ; see e.g. [61] for a recent critical discussion on this aspect. As alluded to above, most sparse estimation methods share this drawback with the parametric methods. However, there are a few sparse methods that are fully data adaptive - we will review one such method called SLIM (sparse learning via iterative minimization) [105] in Section 5.2. These user parameter-free sparse methods have an edge over the parametric ones (which require the selection of C), despite the fact that they cannot be possibly more accurate statistically nor are they necessarily more efficient computationally than the best methods of the parametric class. Additionally, many sparse estimation methods are numerically more reliable than the theoretically more accurate parametric methods (whose global convergence can rarely be guaranteed).

In this chapter we will introduce a semi-parametric/sparse estimation method for the separable model in (5.1) (see also (5.5)). This method will be obtained using a covariance-based fitting criterion, which was apparently never employed before to derive sparse estimation methods, and will be designated by means of the acronym SPICE (**s**emi-**p**arametric/**s**pars**e** iterative **c**ovariance-based **e**stimation). SPICE has several useful features that are shared by very few (if any) sparse estimation methods : i) it is fully data adaptive (i.e., its operation does not require the subtle selection of any user parameters) ; ii) it enjoys global convergence properties ; and iii) its use in the multi-snapshot case

is straightforward. Following the theoretical derivation and analysis of SPICE, we describe the use of this method for the spectral analysis of irregularly-sampled data and make use of a numerical example to illustrate the performance achievable by SPICE in the said application and compare this performance with that of SFLS, IAA and SLIM.

5.2 The competing methods : IAA and SLIM

Let us assume that

$$E(\varepsilon\varepsilon^*) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_N \end{bmatrix} \quad (5.9)$$

and that the phases of $\{s_k\}$ are independently and uniformly distributed in $[0, 2\pi]$. Then the covariance matrix of y has the following expression :

$$\begin{aligned} R &= E(yy^*) = \sum_{k=1}^K |s_k|^2 a_k a_k^* + \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_N \end{bmatrix} \\ &\stackrel{\Delta}{=} A^* P A \end{aligned} \quad (5.10)$$

where

$$A^* = [a_1, \dots, a_K \ I] \stackrel{\Delta}{=} [a_1, \dots, a_{K+N}] \quad (5.11)$$

$$P = \begin{bmatrix} |s_1|^2 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & |s_2|^2 & 0 & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \sigma_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \sigma_N \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & p_2 & 0 & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & p_{K+1} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & \cdots & \cdots & \cdots & \cdots & p_{K+N} \end{bmatrix} \quad (5.12)$$

The assumption that the noise components in different measurements are uncorrelated to one another, which led to (5.9), is quite reasonable in most applications. On the other hand, the assumption that s_k and $s_{\bar{k}}$ are uncorrelated for $k \neq \bar{k}$ does not always hold. However, all methods considered in this chapter are *robust* to this assumption, and thus they work well even in the case when some signal components are correlated to one another, as we explain in the next sub-section.

5.2.1 IAA

Let $p_k(i)$ denote the estimate of p_k at the i -th iteration, and let $R(i)$ be the matrix R made from $\{p_k(i)\}$. Then IAA updates the powers by means of the following iterative process ([122] [96]) :

$$p_k(i+1) = \frac{|a_k^* R^{-1}(i)y|^2}{[a_k^* R^{-1}(i)a_k]^2} \quad k = 1, \dots, K+N \quad (5.13)$$

The initial estimates $\{p_k(0)\}$ can be obtained, for example, using the SFLS method (see (5.6)) :

$$p_k(0) = \frac{|a_k^* y|^2}{\|a_k\|^4} \quad (5.14)$$

The power estimation formula (5.13) has satisfactory properties even in those cases in which the covariance matrix of y does not have the assumed structure due to, for example, coherent signal components (i.e., components with the same phase). To explain why this is so, let us assume that two components, $a_1 s_1$ and $a_2 s_2$, in y are coherent. Then their covariance matrix is not

$$|s_1|^2 a_1 a_1^* + |s_2|^2 a_2 a_2^* \quad (5.15)$$

as assumed in (5.10), but

$$[|s_1|a_1 + |s_2|a_2][|s_1|a_1 + |s_2|a_2]^* \quad (5.16)$$

However this difference between (5.15) and (5.16) does not cause any serious problem to (5.13). Observe that (5.13) can be re-written as (omitting the iteration index for simplicity)

$$p_k = |h_k^* y|^2 \quad ; h_k^* = \frac{a_k^* R^{-1}}{a_k^* R^{-1} a_k} \quad (5.17)$$

The “filter” (or linear combiner) h_k in (5.17) passes without distortion the signal component corresponding to a_k . At the same time it attenuates (or even annihilates, depending on their powers) the signal components corresponding to $a_1 \neq a_k$ and $a_2 \neq a_k$, as it should (see [98] for details on this aspect). For $a_k = a_1$ the filter will attenuate any other component with $a_{\tilde{k}} \neq a_1$, including a_2 (and similarly, for $a_k = a_2$). Note that a similar argument also applies to the methods of SLIM and SPICE which are yet to be discussed : indeed, as we will see, the estimation formulas of these methods comprise a filter that is proportional to the h_k in (5.17). This fact explains why these methods as well are *robust* to the assumed structure of R in (5.10).

5.2.2 SLIM

This method operates under the assumption that $\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma$ (which is a reasonable assumption in some applications). The updated estimates for

SLIM are iteratively obtained as follows ([105] :

$$\begin{aligned} s_k(i+1) &= p_k(i)a_k^*R^{-1}(i)y \quad k = 1, \dots, K \\ p_k(i+1) &= |s_k(i+1)|^2 \quad k = 1, \dots, K \\ \sigma(i+1) &= \frac{1}{N} \left\| y - \sum_{k=1}^K a_k s_k(i+1) \right\|^2 \end{aligned} \quad (5.18)$$

The initial estimates for $\{p_k\}_{k=1}^K$ are obtained as for IAA, whereas $\sigma(0)$ is typically chosen as a small positive number (e.g., $\sigma(0) = 10^{-5}$). Even though derived in the cited papers in a different way, SLIM is similar to the regularized FOCUSS (**focal underdetermined system solver**) algorithm introduced in [38]. The main difference between these two methods consists in the way they estimate σ : SLIM computes the estimate of σ iteratively as in (5.18), while FOCUSS uses a fixed estimate of σ in all iterations that is obtained by one of several possible heuristical methods (see [38]).

Both IAA and SLIM are known to converge *locally* to the minimum value of their corresponding criterion (see the proof of local convergence for IAA in [85] and for SLIM in [105]). However little is known about the global convergence of these algorithms, or in effect about the convergence of their associated sequences.

The main difference between these two algorithms is that IAA is a non-parametric method (which provides a dense power estimate), whereas SLIM is a semi-parametric method (whose result is a sparse power estimate, due to the use of sparsity-inducing parameter priors that lead to an implicit norm constraint similar to the one in (5.8), see [105] for details). In particular, the semi-parametric character of SLIM makes its extension to the multi-snapshot case a bit more difficult than that of IAA for which the extension is more or less straightforward (this difference is due to the fact that for a sparse method, unlike for a dense one, the estimates of $\{s_k\}_{k=1}^K$ for different snapshots should maintain the same sparsity pattern versus the snapshot index) ; we refer to [85] for details on these extensions.

5.3 The proposed method : SPICE

The following is a weighted covariance fitting criterion that can be used for the purpose of parameter estimation (see, e.g., [75] [54] and the references therein ; also see [34] and [12]) :

$$f = \|R^{-1/2}(yy^* - R)\|^2 \quad (5.19)$$

where $\|\cdot\|$ denotes the Frobenius norm for matrices, and $R^{-1/2}$ is the Hermitian positive definite square root of R^{-1} . Admittedly, the use of (5.19) makes more sense in the multi-snapshot case than in the single snapshot one (in which

the sample covariance matrix is just yy^* , as used in (5.19)), but the minimization of f can yield satisfactory estimates even in the latter case (as we explain later on in this section). A simple calculation shows that :

$$f = -2\|y\|^2 + \|y\|^2 y^* R^{-1} y + \text{tr}(R) \quad (5.20)$$

where (see (5.10)-(5.12))

$$\text{tr}(R) = E(\|y\|^2) = \sum_{k=1}^{K+N} \|a_k\|^2 p_k \quad (5.21)$$

(and where E is the expectation operator). It follows from (5.20) and (5.21) that the minimization of f is equivalent to the minimization of the function :

$$g = y^* R^{-1} y + \sum_{k=1}^{K+N} w_k p_k; w_k = \frac{\|a_k\|^2}{\|y\|^2} \quad (5.22)$$

As shown next, this is a convex problem.

5.3.1 Convexity of the problem

The following equivalences can be readily verified :

$$\begin{aligned} \min_{\{p_k\}} g &\Leftrightarrow \min_{x, \{p_k \geq 0\}} x + \sum_{k=1}^{K+N} w_k p_k \quad \text{s.t. } x \geq y^* R^{-1} y \Leftrightarrow \\ &\min_{x, \{p_k \geq 0\}} x + \sum_{k=1}^{K+N} w_k p_k \quad \text{s.t. } \begin{bmatrix} x & y^* \\ y & R \end{bmatrix} \geq 0 \end{aligned} \quad (5.23)$$

The minimization problem in (5.23) (where $R = A^* P A$, see (5.10)) is a semi-definite program (SDP) [16] which is well known to be convex. The convexity of the original problem therefore follows.

There are several well-documented software packages for solving an SDP such as (5.23) (see, e.g., [104]). However SDP solvers are in general rather computationally intensive (as an example, using such a solver for (5.23) with $N = 50$ and $K = 1000$ takes about 1 hour on a relatively powerful PC). Consequently, we do not recommend solving the SDP in (5.23) as the preferred method for estimating the powers $\{p_k\}$, and suggest a different line of attack in the next sub-section.

5.3.2 Derivation of SPICE

The literature on optimal experiment design contains a host of results on the minimization of functions of the form of our $y^* R^{-1} y$, under the constraints that $p_k \geq 0$ and $\sum_{k=1}^{K+N} w_k p_k = 1$; see, e.g., [125] [79] and the many references therein.

In order to make use of these results, particularly those of [125] (which inspired the derivation of SPICE below), we consider a reformulation of the problem introduced above. Specifically, it follows from (5.21) that $\|y\|^2$ is an unbiased and consistent (in N) estimate of $\sum_{k=1}^{K+N} \|a_k\|^2 p_k$. Based on this observation we will estimate the powers by solving the following linearly-constrained minimization problem, instead of minimizing the g in (5.22):

$$\min_{\{p_k \geq 0\}} y^* R^{-1} y \quad \text{s.t. } \sum_{k=1}^{K+N} w_k p_k = 1 \quad (5.24)$$

It can be shown ([34]) that the problems (5.22) and (5.24) are equivalent in the sense that the solution to (5.22) is a scaled version of the solution of (5.24). Note that (5.24) is also a convex problem, see the previous sub-section. Additionally note that the constraint in (5.24) is of the (weighted) ℓ_1 -norm type, and therefore it is expected to be sparsity inducing for the solution to (5.24) (see, e.g., [107] [61] [38]).

Let $Q \in \mathbb{C}^{(K+N) \times N}$ be such that $Q^* A = I$, and let (assuming $\{p_k > 0\}$) to simplify the explanations; to accommodate the case of $p_k = 0$, for some values of k , a pseudo-inverse has to be used instead of the matrix inverse below, see [125]):

$$h = y^* Q^* P^{-1} Q y \quad (5.25)$$

The minimization of h with respect to Q , under the constraint $Q^* A = I$, yields (see 7.5 for a proof) :

$$Q_0 = P A R^{-1} \quad (5.26)$$

and

$$\begin{aligned} h_0 &\stackrel{\Delta}{=} h |_{Q=Q_0} = y^* R^{-1} y \\ &= \text{the original objective in (5.24)} \end{aligned} \quad (5.27)$$

The important consequence of this result is that the minimization of h with respect to $\{p_k\}$ and Q (s.t. $Q^* A = I$) results in the same $\{p_k\}$ as the minimization of (5.24). The usefulness of this observation lies in the fact that the minimization of the augmented function h can be conveniently done by means of a cyclic (aka alternating) algorithm that consists of iterating the following steps until convergence :

Step 0. Compute initial estimates of $\{p_k\}$, e.g. by using (5.14).

Step 1. With $\{p_k\}$ fixed at their most recent estimates, minimize h with respect to Q , s.t. $Q^* A = I$. The minimizing matrix Q is given by (5.26).

Step 2. With Q fixed at its most recent value, minimize h with respect to $\{p_k \geq 0\}$ s.t. $\sum_{k=1}^{K+N} w_k p_k = 1$. The solution of this step can also be obtained in closed form, as detailed below.

Let

$$Q(i) = P(i) A R^{-1}(i) \quad (5.28)$$

be the Q matrix in (5.26) at the i -th iteration of the algorithm, and let

$$\beta(i) = Q(i)y = P(i)AR^{-1}(i)y \quad (5.29)$$

Then the minimization problem that needs to be solved in Step 2 of the above algorithm, at its i -th iteration, is :

$$\min_{\{p_k \geq 0\}} \sum_{k=1}^{K+N} \frac{|\beta_k(i)|^2}{p_k} \quad \text{s.t. } \sum_{k=1}^{K+N} w_k p_k = 1 \quad (5.30)$$

where $\beta_k(i)$ is the k -th element of the vector $\beta(i)$ viz.

$$\beta_k(i) = p_k(i)a_k^*R^{-1}(i)y \quad (5.31)$$

By the Cauchy-Schwartz inequality we have that :

$$\begin{aligned} \left[\sum_{k=1}^{K+N} w_k^{1/2} |\beta_k(i)| \right]^2 &= \left[\sum_{k=1}^{K+N} \frac{|\beta_k(i)|}{p_k^{1/2}} w_k^{1/2} p_k^{1/2} \right]^2 \\ &\leq \left[\sum_{k=1}^{K+N} \frac{|\beta_k(i)|^2}{p_k} \right] \left[\sum_{k=1}^{K+N} w_k p_k \right] \\ &= \left[\sum_{k=1}^{K+N} \frac{|\beta_k(i)|^2}{p_k} \right] \end{aligned} \quad (5.32)$$

It follows immediately from (5.32) that the solution to (5.30), and hence the solution to Step 2 of the cyclic algorithm, is given by :

$$p_k(i+1) = \frac{|\beta_k(i)|}{w_k^{1/2} \sum_{l=1}^{K+N} w_l^{1/2} |\beta_l(i)|} \quad (5.33)$$

Inserting (5.31) into (5.33) we get the following compact expression for the iterative *power updates of the SPICE algorithm* :

$$\begin{aligned} p_k(i+1) &= p_k(i) \frac{|a_k^* R^{-1}(i)y|}{w_k^{1/2} \rho(i)} \\ \rho(i) &= \sum_{l=1}^{K+N} w_l^{1/2} p_l(i) |a_l^* R^{-1}(i)y| \end{aligned} \quad (5.34)$$

Interestingly, the above equation has the same multiplicative form as the power update formula for SLIM, see (5.18), whereas the updating equation for IAA has a different form. Furthermore, the updating formulas for all three methods depend on $|a_k^* R^{-1}(i)y|$, although the power to which this quantity is raised depends on the method (1st power for SPICE and 2nd power for IAA and SLIM).

With regard to implementation, SPICE and SLIM can be implemented quite efficiently by first computing $z(i) = R^{-1}(i)y$ (possibly by means of a conjugate-gradient algorithm, see e.g. [27]) and then evaluating the scalar products $a_k^* z(i)$ (for $k = 1, \dots, K+N$). The implementation of IAA, on the other hand, requires comparatively more computations due to the need for evaluating the denominator in (5.13) for $k = 1, \dots, K+N$ (the reader is reminded that $K \gg N$ usually). For easy reference, the SPICE algorithm is summarized in Table 5.1.

Step	Computation/operation	Eq. no.
0) Initialization	$p_k = a_k^* y ^2 / \ a_k\ ^4 \quad (k = 1, \dots, K+N)$	(5.14)
1) Computation of R	$R = \sum_{k=1}^{K+N} p_k a_k a_k^*$	(5.10)
2) Power update	$z = R^{-1} y$ $w_k^{1/2} = \ a_k\ / \ y\ \quad (k = 1, \dots, K+N)$ $r_k = a_k^* z \quad (k = 1, \dots, K+N)$ $\rho = \sum_{l=1}^{K+N} w_l^{1/2} p_l r_l$ $p_k \leftarrow p_k r_k / w_k^{1/2} \rho \quad (k = 1, \dots, K+N)$	(5.34)
Iteration : iterate steps 1 and 2 until convergence.		

Table 5.1. The SPICE algorithm.

5.3.3 Some theoretical properties

Because SPICE monotonically decreases the objective function (due to its cyclic operation) and since the minimization problem it solves is convex, we can expect that the algorithm has global convergence properties. In effect, it can be shown that under reasonably weak conditions (essentially requiring that $\{p_k(0) > 0\}$ and that the matrix $R(i)$ remains positive definite as i increases), *the limit points of SPICE power sequence, (5.34), are global minimizers of $y^*R^{-1}y$ subject to the constraints in (5.24)* [125]. In other words, the SPICE algorithm is globally convergent for any initial value that belongs to the interior of the set $\{p_k \geq 0\}$, which is usually the case (e.g., for (5.14)). Despite the said initialization, the limit points of the algorithm tend often to be on the boundary of the above set, a fact in agreement with the previously made observation that SPICE is a sparse estimation method (see the comments following (5.24)).

The theoretical analysis of the global minimum points of $y^*R^{-1}y$ appears to be difficult. Empirical experience with the SPICE algorithm suggests that typically the locations of the dominant peaks of the true power spectrum are well determined, but also that the corresponding power values require some form of correction. To understand this type of behavior, we consider the noise free case in which (see (5.1) with $\varepsilon = 0$) :

$$y = Bb \quad (5.35)$$

for some vector $b \in \mathbb{C}^{C \times 1}$ and a matrix $B \in \mathbb{C}^{N \times C}$ that is a block of A^* . Furthermore, let $\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma$ (the true value of which is $\sigma = 0$ in the present case), let all the other powers in P be zero except for those corresponding to the columns of B , and let

$$D^2 = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & d_C \end{bmatrix} \quad (5.36)$$

be the diagonal matrix corresponding to the non-zero powers. Then, by the matrix inversion lemma,

$$R^{-1} = [(BD)(DB^*) + \sigma I]^{-1} = \frac{1}{\sigma}I - \frac{1}{\sigma}BD(\sigma I + DB^*BD)^{-1}DB^* \quad (5.37)$$

which implies that :

$$\begin{aligned} y^*R^{-1}y &= \frac{1}{\sigma}b^* [B^*B - B^*BD(\sigma I + DB^*BD)^{-1}DB^*B] b \\ &= \frac{1}{\sigma}b^*B^*BD(\sigma I + DB^*BD)^{-1} [(\sigma I + DB^*BD)D^{-1} - DB^*B] b \\ &= b^*B^*BD(\sigma I + DB^*BD)^{-1}D^{-1}b \\ &= b^* [D(\sigma I + DB^*BD)(B^*BD)^{-1}]^{-1} b \\ &= b^* [\sigma(B^*B)^{-1} + D^2] b \end{aligned} \quad (5.38)$$

In many cases of interest $(B^*B)^{-1} \rightarrow 0$ as $N \rightarrow \infty$; for example this happens in the spectral analysis application (see Section 5.4), in which $\|a_k\|^2 = N$ - this is assumed in the following. So let us assume for simplicity that the term $\sigma(B^*B)^{-1}$ in (5.38) can be neglected. Then the minimization problem in (5.24) becomes, approximately,

$$\min_{\{d_k\}, \sigma} \sum_{k=1}^C \frac{|b_k|^2}{d_k} \quad \text{s.t. } \sum_{k=1}^C d_k + \sigma = \frac{\|y\|^2}{N} \quad (5.39)$$

the solution to which is (by a calculation similar to that in (5.32)) :

$$\sigma = 0^+ \quad (\sigma \text{ must satisfy } \sigma > 0 \text{ for } R^{-1} \text{ to exist}) \quad (5.40)$$

$$d_k = |b_k| \rho \quad (5.41)$$

where

$$\rho = \frac{\|y\|^2}{N \left(\sum_{k=1}^C |b_k| \right)} \quad (5.42)$$

is a normalizing constant. It follows from (5.41) that the powers (at the true locations) which minimize the SPICE criterion are proportional to the square root of the true powers :

$$d_k = \rho (|b_k|^2)^{1/2} \quad (5.43)$$

Note, however, that for $C = 1$ we have $\|y\|^2 = N|b_1|^2$ and hence the power is correctly determined in this case : $d_1 = |b_1|^2$.

Another way to realize that (5.43) is true is based on (5.29) and (5.33). For given P (and hence R), the $\{\beta_k\}$ in (5.29) (we omit the index i to simplify the notation) are readily recognized to be estimates of the signal amplitudes as well as of the noise term in (5.5) ; now the $\{p_k\}$ in (5.33) are proportional to $\{|\beta_k|\}$: therefore $\{p_k\}$ obtained from SPICE must be (scaled) estimates of the square root of the powers, as $\{|\beta_k|\}$ are. It also follows from this discussion that, whenever accurate determination of the heights of the power peaks (not only of their locations) is required, we can proceed in the following way. First we use the $\{|\beta_k|^2\}$ obtained from SPICE (at convergence) to re-estimate the powers :

$$\hat{p}_k = |\beta_k|^2 \quad k = 1, \dots, K+N \quad (5.44)$$

Note that the $\{\beta_k\}$ obtained from (5.29) would be the best linear unbiased estimates if P used in (5.29) were the true power matrix (or a scaled version thereof). However, as explained above, the powers produced directly by SPICE do not satisfy this condition, which means that the accuracy of (5.44) might not be satisfactory. To improve the estimation performance of (5.44) we use it three times to re-estimate the powers : each time we use the most recent power estimates to build P , compute $\{\beta_k\}$ with (5.29), and then obtain enhanced power estimates via (5.44).

Remark 4 Once the locations of the dominant power peaks are estimated with SPICE, their heights can be determined by means of several other methods (see, e.g., [95]), besides the one outlined above. Of these possible methods, we have tested the one based on multiple-frequency least squares (MFLS) and found out that its performance in the spectral analysis application presented in Section 5.4 was quite satisfactory. However, we have chosen to omit the details on MFLS and to focus on the method based on (5.29) and (5.44), because the latter method is inherently intertwined with SPICE, unlike MFLS. ■

Before concluding this discussion, we remark on the fact that in the previous analysis we let $R = BD^2B + \sigma I$ where D and σ were unknown, but B was the matrix made from the true signal vectors $\{a(\tilde{\omega}_c)\}$ present in y (see (5.1)). An interesting question is what happens if we replace B in R by a matrix that is also unknown, let us say $X \in \mathbb{C}^{N \times C}$: will the minimization of the SPICE criterion yield the true solution $X = B$, or at least a close approximation of it? As shown in 7.6 the answer to this question is positive, therefore lending support to the previously asserted fact that the locations of the true power spectrum peaks are well determined by SPICE.

5.3.4 Some extensions

In some applications the powers $\{p_k\}$ satisfy certain known linear relationships. For example, we might know that the *noise variance in the different measurements is constant*, i.e.

$$\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma \quad (5.45)$$

The extension of SPICE to include the above information is immediate. We only need to observe from (5.28) and (5.29) that, under (5.45), the term that multiplies σ in (5.30) is :

$$\sum_{k=K+1}^{K+N} |\beta_k(i)|^2 = \sigma^2(i) \|R^{-1}(i)y\|^2 \quad (5.46)$$

Consequently the updating formulas for SPICE become :

$$p_k(i+1) = p_k(i) \frac{|a_k^* R^{-1}(i)y|}{w_k^{1/2} \rho(i)} \quad (5.47)$$

$$\sigma(i+1) = \sigma(i) \frac{\|R^{-1}(i)y\|}{v^{1/2} \rho(i)} \quad (5.48)$$

where

$$v = \sum_{k=K+1}^{K+N} w_k \quad (5.49)$$

and

$$\rho(i) = \sum_{l=1}^K w_l^{1/2} p_l(i) |a_l^* R^{-1}(i)y| + v^{1/2} \sigma(i) \|R^{-1}(i)y\| \quad (5.50)$$

Similarly to the discussion in the previous sub-section, whenever accurate estimation of the powers is necessary, we can use (5.29) and (5.44) to obtain refined estimates of $\{p_k\}_{k=1}^K$ and of σ :

$$\hat{p}_k = |\beta_k|^2 \quad k = 1, \dots, K \quad \hat{\sigma} = \frac{1}{N} \sum_{k=K+1}^{K+N} |\beta_k|^2 \quad (5.51)$$

As outlined before we repeat three times the calculation in (5.51), each time using the latest estimates of $\{p_k\}_{k=1}^K$ and σ to determine $\{\beta_k\}_{k=1}^{K+N}$ via (5.29). This is the SPICE power estimation formula that will be used in the numerical study in Section 5.4.

In other applications the power spectrum is known to possess certain symmetries. A prime example of this situation is the *processing of real-valued data*, in which case constraints of the type

$$p_k = p_{\bar{k}} \quad (5.52)$$

for certain values of $k \neq \bar{k}$ are known to hold. Modifying SPICE to take (5.52) into account can be done as indicated above for the similar constraint in (5.45), and therefore we omit the details of this extension.

In the rest of this sub-section we explain how to extend SPICE to the case of *multiple snapshots*. Let $Y \in \mathbb{C}^{N \times M}$ denote the matrix whose columns are the M available data snapshots, and let

$$\hat{R} = \frac{1}{M} YY^* \quad (5.53)$$

The weighted covariance fitting criterion associated with (5.53) is given by (compare with (5.19)):

$$f = \|R^{-1/2}(\hat{R} - R)\|^2 = \text{tr}(\hat{R}R^{-1}\hat{R}) + \text{tr}(R) - 2\text{tr}(\hat{R}) \quad (5.54)$$

where

$$\text{tr}(R) = E(\text{tr}(\hat{R})) = \sum_{k=1}^{K+N} \|a_k\|^2 p_k \quad (5.55)$$

An extended version of the SPICE estimation criterion, see (5.24), follows easily from the above equation:

$$\min_{\{p_k \geq 0\}} \text{tr}(\hat{R}R^{-1}\hat{R}) \quad \text{s.t. } \sum_{k=1}^{K+N} w_k p_k = 1 \quad (5.56)$$

where

$$w_k = \frac{\|a_k\|^2}{\text{tr}(R)} \quad (5.57)$$

The derivation of the SPICE algorithm for the extended problem in (5.56) parallels that in Section 5.3.2 for the case of $M = 1$. Consider the function :

$$h = \text{tr}(\hat{R}Q^*P^{-1}Q\hat{R}) \quad (5.58)$$

For fixed P , the matrix Q that minimizes (5.58) (s.t. $Q^*A = I$) is still given by (5.26) :

$$Q_0 = PAR^{-1} \quad (5.59)$$

and it is still true that

$$\begin{aligned} h|_{Q=Q_0} &= \text{tr}(\hat{R}R^{-1}\hat{R}) \\ &= \text{the original objective in (5.56)} \end{aligned} \quad (5.60)$$

Next, observe that the function in (5.58) can be re-written (for fixed Q) as :

$$h = \sum_{k=1}^{K+N} \frac{|\beta_k|^2}{p_k} \quad (5.61)$$

where now $\{|\beta_k|^2\}$ are the diagonal elements of the matrix $Q\hat{R}^2Q^*$, i.e.

$$|\beta_k|^2 = [Q\hat{R}^2Q^*]_{kk} = \frac{1}{M^2} [QY(Y^*Y)Y^*Q^*]_{kk} \quad (5.62)$$

Owing to the perfect analogy between (5.61) and the equation (5.30) corresponding to the case $M = 1$, we conclude that the extended algorithm has exactly the same form as the basic SPICE algorithm, the only modification being the different expressions for $\{w_k\}$ and for $\{|\beta_k|^2\}$ (see (5.57) and (5.62) above). Note also that whenever $M > N$ the $\{p_k\}$ obtained from SPICE are usually accurate estimates of the true powers and therefore, unlike in the case of $M = 1$, we can use them directly as power estimates - which is what we will do in the numerical study of the next section.

5.4 Application to spectral analysis and numerical performance study

Let $y_k = y(t_k)$, $k = 1, \dots, N$, be the k -th sample in a set of N measurements performed at possibly irregularly-spaced times $\{t_k\}$. In spectral analysis applications the vectors $\{a(\omega_k)\}$ correspond to sinusoidal components :

$$a(\omega_k) = \begin{bmatrix} e^{i\omega_k t_1} \\ \vdots \\ e^{i\omega_k t_N} \end{bmatrix} \quad (5.63)$$

(observe that in this case $\|a(\omega_k)\|^2 = N$ is a constant ; see (5.21), (5.22) for equations in which $\{\|a(\omega_k)\|^2\}$ appear). The interval for the angular frequency, viz. Ω , in which one can conduct spectral analysis without any alias-

ing problem, can be determined from the so-called spectral window, $\left| \frac{1}{N} \sum_{k=1}^N e^{i\omega_k t_k} \right|^2$,

as the largest range for ω in which the only peak with height equal (or close) to 1 is at $\omega = 0$ [31]. Let $\Omega = [-\omega_{max}, \omega_{max}]$ denote the so-obtained interval, or the sub-interval of it that is of interest. We use a uniform grid $\{\omega_k\}_{k=1}^K$ to cover Ω , with a step equal to $\frac{2\pi}{5(t_N - t_1)}$ (note that $\frac{2\pi}{(t_N - t_1)}$ is the best expected resolution in the class of non-parametric methods [98] ; because we can hope for a better resolution in the sparse-estimation method class, we choose a 5 times smaller step for the frequency grid).

We will consider $N = 100$ and the sampling pattern shown in Fig 5.1a. This sampling pattern, which will be fixed in the simulation runs, mimics the type of patterns encountered in certain applications of spectral analysis in astronomy (see [63] [5]) where data collection depends on many factors and therefore is typically performed at rather irregular time intervals. For the sampling pattern in question we selected $\omega_{max} = \pi$. The spectral window is shown in Fig 5.1b from which one can see that the only peak with height equal (or close) to 1 in the frequency range of $[-\pi, \pi]$ is at $\omega = 0$, as required. The grid size K is chosen as $K = 5(t_N - t_1) = 1000$.

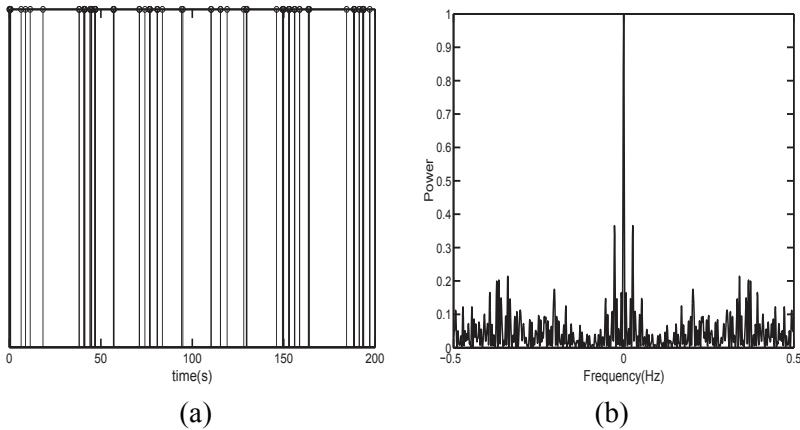


Figure 5.1. a) The sampling pattern, for $N = 100$, mimicking a real-life case in astronomy [63]. b) The corresponding spectral window

The data samples were simulated using equations (5.1) and (5.63) with $C = 3$, $\tilde{\omega}_1 = 2\pi \times 0.3100$, $\tilde{\omega}_2 = 2\pi \times 0.3150$, $\tilde{\omega}_3 = 2\pi \times 0.1450$, $\tilde{s}_1 = 10e^{i\varphi_1}$, $\tilde{s}_2 = 10e^{i\varphi_2}$, and $\tilde{s}_3 = 3e^{i\varphi_3}$, where the phases $\{\varphi_k\}_{k=1}^3$ were independently and uniformly distributed in $[0, 2\pi]$. The disturbance term, ε , was normal white noise with mean zero and variance σ ; the noise variance will be varied to control the signal-to-noise ratio defined as :

$$\text{SNR} = 10 \log\left(\frac{100}{\sigma}\right) = 20 - 10 \log \sigma \quad [\text{dB}]$$

The following methods will be used for spectrum/parameter estimation :

M_1 : SFLS.

M_2 : IAA.

M_3 : SLIM.

M_4 : SPICE_{SS} = SPICE - single snapshot with $\sigma_1 = \dots = \sigma_N$.

M_5 : SPICE_{MS} = SPICE - multi-snapshot with $\sigma_1 = \dots = \sigma_N$.

We will not consider SPICE with different $\{\sigma_k\}$ as it is unlikely to provide better performance in the present case in which the noise elements have the same variance. However, note that in some applications (such as astronomy) the noises affecting different measurements can indeed have different variances, in which case SPICE with unconstrained $\{\sigma_k\}$ should be used. Regarding M_5 , we note that replicated measurements are rarely available in spectral analysis applications. However in other applications, such as in array processing, they are a common occurrence and this is the reason for including M_5 in the present comparison. As the number of snapshots increases, the accuracy of M_5 improves significantly ; we use $M = 1000$ in the following. We note in this context that IAA, and probably SLIM too, can also be extended to the multi-snapshot case ([85]). However, the said extensions are not as well motivated as the proposed extension of SPICE, and on that basis they will not be considered in this numerical study. The stopping criterion for the iterative methods, namely, IAA, SLIM and SPICE, is $\frac{\|p^{i+1} - p^i\|}{\|p^i\|} < 10^{-4}$, where p^i denotes the estimate of the power vector at the i -th iteration. The average number of iterations and the average computation time (on a 2.26GHz, 4GB PC) for these methods in the present example (with SNR = 10dB) are as follows : IAA - 940 iterations and 158 sec, SLIM - 20 iterations and 3 sec and SPICE_{SS} - 180 iterations and 20 sec.

In Fig 5.2 we display, in a superimposed manner, the power spectrum estimates obtained with $M_1 - M_5$ in 100 Monte-Carlo (MC) runs, for SNR = 10dB. It can be inferred from the plots in this figure that SFLS suffers from heavy leakage problems ; as a result the peak at $\tilde{\omega}_3$ is buried completely in "clutter", and the frequencies $\tilde{\omega}_1$ and $\tilde{\omega}_2$ are estimated with a bias that can be seen clearly from the insert. IAA, on the other hand, resolves the two closely spaced peaks without any bias but misses the weaker component at $\tilde{\omega}_3$ possibly due to ill-conditioning of the matrix R caused by irregular sampling (note that IAA, in general, works reasonably well for uniform sampling schemes). The semi-parametric method SLIM yields a sparse spectrum but misses the true peaks in some MC runs, and also over-estimates the powers in some realizations (note that some of the peaks at $\tilde{\omega}_1$ and $\tilde{\omega}_2$ in Fig 5.2c are larger than 200 (see the insert) ; however we cut them off at 200 in order to be able to use the same scale for all plots in Fig 2). SPICE_{SS}, on the other hand, locates all three peaks in most MC runs and also gives more accurate power estimates. Finally SPICE_{MS}, yields a nearly ideal spectrum.

In Figs 3-7 we show the histograms, obtained from 100 MC runs, of the frequency and power estimates obtained with $M_1 - M_5$ for two different values of SNR. For all methods the frequency estimates are obtained as the locations

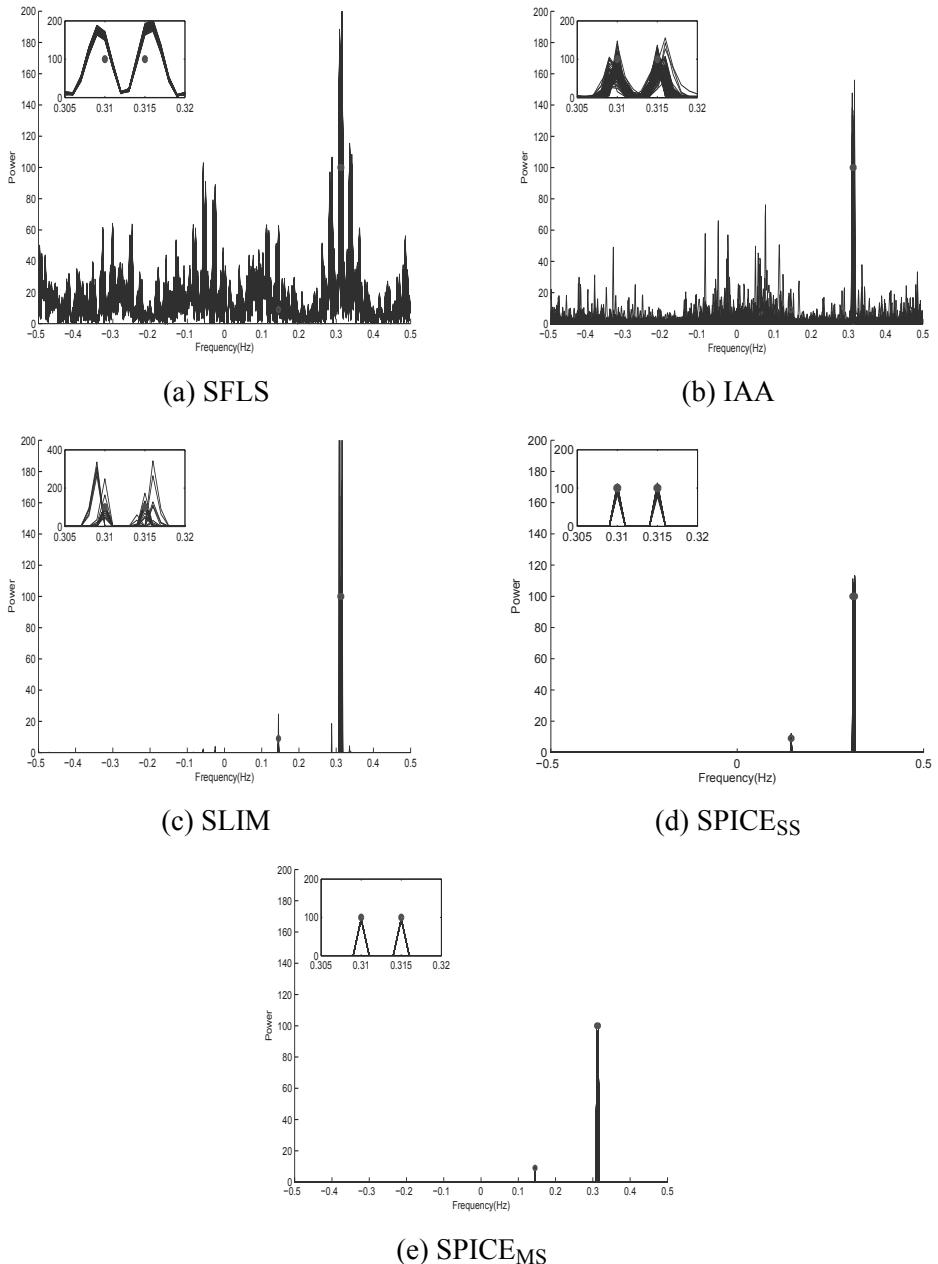


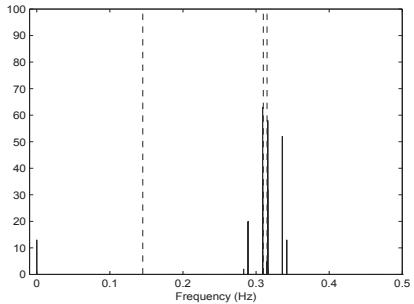
Figure 5.2. Superimposed spectra obtained with the methods under consideration in 100 Monte Carlo runs, SNR = 10dB. The circles in the plots indicate the true value of the powers.

of the three largest peaks of the corresponding estimated spectrum and the power estimates are computed at the estimated frequencies. In the histogram for power estimates, the values were saturated at 300 (i.e. larger values are not shown to focus on the power range of interest). Similarly, in the histogram for frequency estimates, the values were saturated at 0 (i.e. negative values are not shown). As expected, SFLS works poorly at low SNR and even at SNR = 20dB it fails to locate the component at $\tilde{\omega}_3$ (instead, in all 100 MC runs, it picked a spurious peak to the right of $\tilde{\omega}_2$, which is an artifact due to the sampling scheme). Note also that the SFLS power estimates at $\tilde{\omega}_1$ and $\tilde{\omega}_2$ are significantly overestimated. IAA gives poor frequency and power estimates at the low SNR, which is primarily due to the ill-conditioning of R caused by the irregular sampling scheme; however, it gives satisfactory frequency and power estimates at the SNR value of 20dB. The accuracy of the frequency and power estimates obtained with SLIM is relatively poor at both SNR values considered ; this is mainly due to the fact that SLIM gives a too sparse spectrum and hence fails to detect some of the components present in the data. The frequency estimation performance of SPICE_{SS} is superior to that of SFLS, IAA and SLIM, at both low and high SNR values. The SPICE_{SS} power estimates are also reasonable, despite the fact that they are somewhat biased downwards at $\tilde{\omega}_1$ and $\tilde{\omega}_2$. Finally SPICE_{MS} gives very accurate frequency estimates and precise power estimates for both SNR values.

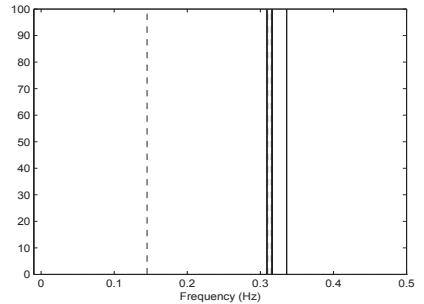
5.5 Concluding remarks

The SPICE (semi-parametric/sparse iterative covariance-based estimation) method introduced in this chapter enjoys global convergence properties, is user parameter free, can be easily used in the multi-snapshot case, and has a small computational complexity. There are very few (if any) other sparse estimation methods that share all these useful features. In our opinion the capability of SPICE to operate in noisy scenarios without requiring prior information about the noise variance or the sparsity index of the parameter vector as well as its simple extension to multi-snapshot situations are particularly useful characteristics. Regarding the statistical properties of the parameter estimates provided by SPICE, the proposed method was shown to outperform two competing methods in a numerical spectral analysis application.

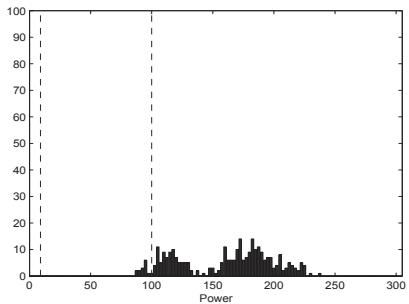
Separable models appear in many branches of science and engineering, and we are planning to investigate the use of SPICE in several other applications, besides spectral analysis. Fully understanding the theoretical properties of SPICE also requires some additional research work. For example, as mentioned briefly in the footnote to the discussion in Section 5.3.3, extensions of certain representation results for solutions to SPICE-type optimization problems are likely to shed further light on the computational and statistical properties of SPICE as well as on its relationship to the group of sparse estimation



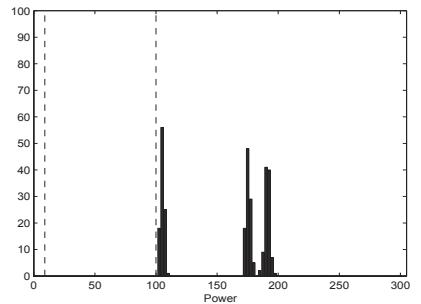
(a) SNR = 0dB



(b) SNR = 20dB

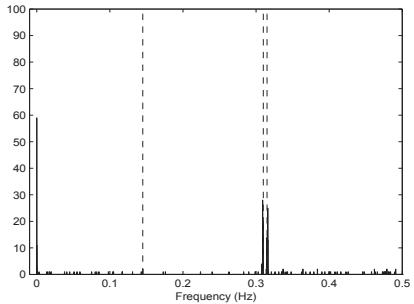


(c) SNR= 0dB

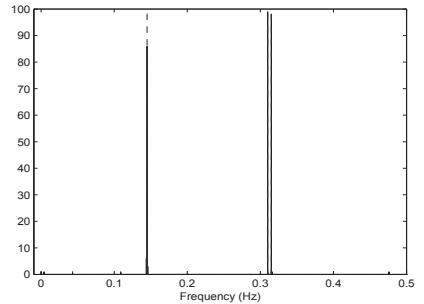


(d) SNR = 20dB

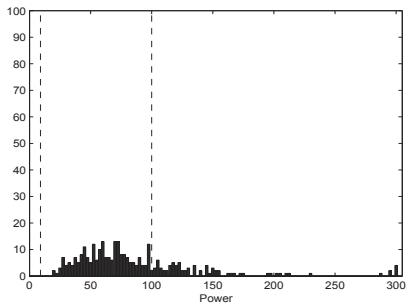
Figure 5.3. Histograms of (a)-(b) frequency estimates and (c)-(d) power estimates for SFLS at two SNR values, obtained from 100 Monte Carlo runs. The dashed lines show the true frequencies and the true powers.



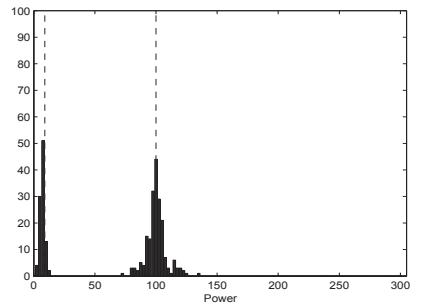
(a) SNR = 0dB



(b) SNR = 20dB

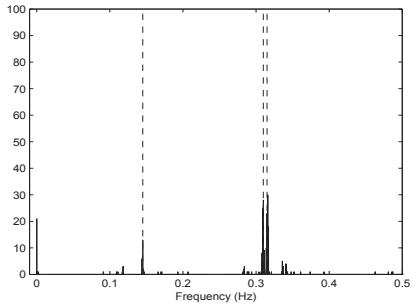


(c) SNR = 0dB

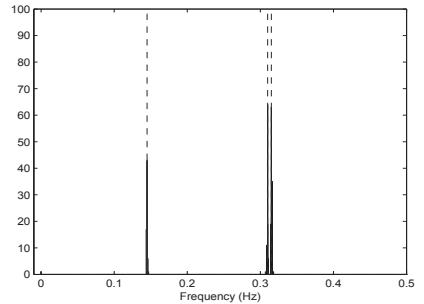


(d) SNR = 20dB

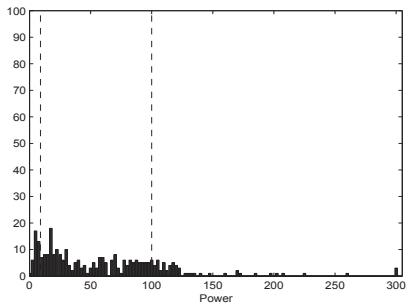
Figure 5.4. Histograms of (a)-(b) frequency estimates and (c)-(d) power estimates for IAA at two SNR values, obtained from 100 Monte Carlo runs. The dashed lines show the true frequencies and the true powers.



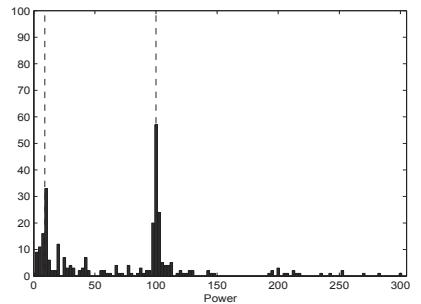
(a) SNR = 0dB



(b) SNR = 20dB

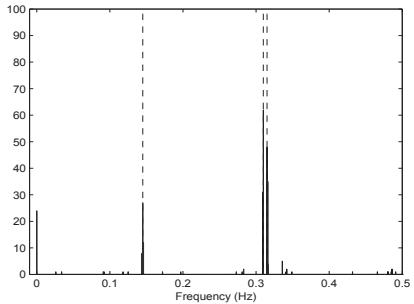


(c) SNR = 0dB

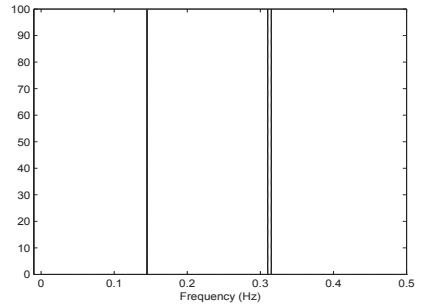


(d) SNR = 20dB

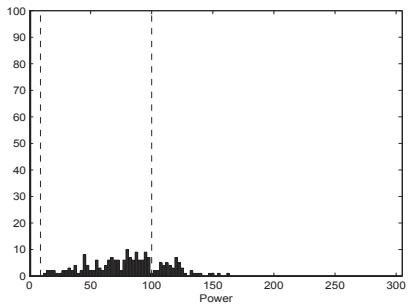
Figure 5.5. Histograms of (a)-(b) frequency estimates and (c)-(d) power estimates for SLIM at two SNR values, obtained from 100 Monte Carlo runs. The dashed lines show the true frequencies and the true powers.



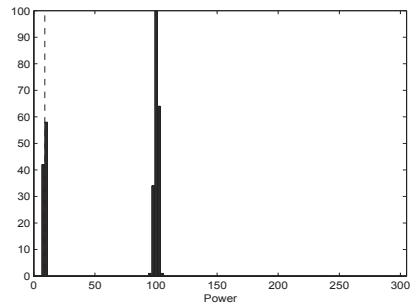
(a) SNR = 0dB



(b) SNR = 20dB

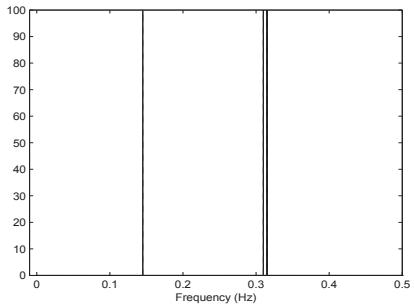


(c) SNR = 0dB

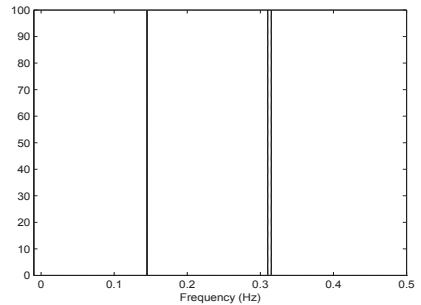


(d) SNR = 20dB

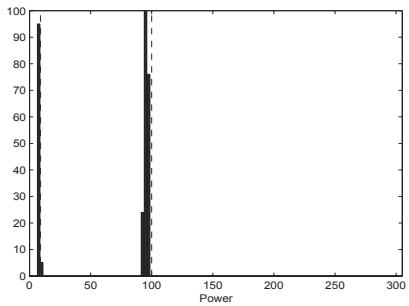
Figure 5.6. Histograms of (a)-(b) frequency estimates and (c)-(d) power estimates for SPICE_{SS} at two SNR values, obtained from 100 Monte Carlo runs. The dashed lines show the true frequencies and the true powers.



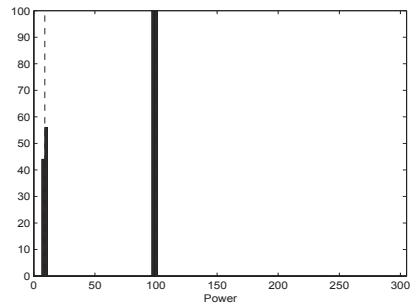
(a) SNR = 0dB



(b) SNR = 20dB



(c) SNR = 0dB



(d) SNR = 20dB

Figure 5.7. Histograms of (a)-(b) frequency estimates and (c)-(d) power estimates for SPICE_{MS} at two SNR values, obtained from 100 Monte Carlo runs. The dashed lines show the true frequencies and the true powers.

methods based on ℓ_1 -norm minimization principles. We leave working out such theoretical extensions and providing a more detailed analysis of SPICE to a possible future publication.

5.6 Proof of (5.26) and (5.27)

To prove (5.26) we need to show that

$$Q_0^* P^{-1} Q_0 = R^{-1} \leq Q^* P^{-1} Q \quad (5.64)$$

for any Q that satisfies $Q^* A = I$. However (5.64) is equivalent to :

$$\begin{aligned} \begin{bmatrix} Q^* P^{-1} Q & I \\ I & A^* P A \end{bmatrix} \geq 0 &\Leftrightarrow \\ \begin{bmatrix} Q^* P^{-1} Q & Q^* A \\ A^* Q & A^* P A \end{bmatrix} &= \begin{bmatrix} Q^* & 0 \\ 0 & A^* \end{bmatrix} \begin{bmatrix} P^{-1/2} \\ P^{1/2} \end{bmatrix} \begin{bmatrix} P^{-1/2} & P^{1/2} \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & A \end{bmatrix} \geq 0 \end{aligned} \quad (5.65)$$

and (5.65) evidently holds true. Therefore (5.26) is proved, and (5.27) follows by substitution.

5.7 On the minimizers of the SPICE criterion in the noise-free case

Let y be given by (5.35), and let

$$R = X D^2 X^* + \sigma I \quad (5.66)$$

where D^2 is as in (5.36), and $X \in \mathbb{C}^{N \times C}$ is a matrix, made from columns of A^* , that has full column rank. A straightforward calculation yields :

$$\begin{aligned} R^{-1} &= \frac{1}{\sigma} I - \frac{1}{\sigma} X D (\sigma I + D X^* X D)^{-1} D X^* \\ &= \frac{1}{\sigma} [I - X D (D X^* X D)^{-1} D X^*] \\ &\quad + \frac{1}{\sigma} [X D (D X^* X D)^{-1} D X^* - X D (\sigma I + D X^* X D)^{-1} D X^*] \\ &= \frac{1}{\sigma} [I - X (X^* X)^{-1} X^*] + \frac{1}{\sigma} X D (D X^* X D)^{-1} [(\sigma I + D X^* X D) \\ &\quad - D X^* X D] (\sigma I + D X^* X D)^{-1} D X^* \\ &= \frac{1}{\sigma} \Pi + X (X^* X)^{-1} D^{-2} (\sigma D^{-2} + X^* X)^{-1} X^* \end{aligned} \quad (5.67)$$

where Π is the orthogonal projector matrix onto the null space of X^* . Similarly to what we have done in Section 5.3.3 we can neglect the term σD^{-2} in (5.68), as it is typically much smaller than $X^* X$. Therefore we have (approximately) :

$$y^* R^{-1} y = \frac{1}{\sigma} b^* B^* \Pi B b + b^* B^* X (X^* X)^{-1} D^{-2} (X^* X)^{-1} X^* B b \quad (5.68)$$

The minimization of (5.68) with respect to $\{d_k\}$, s.t. $\sum_{k=1}^C d_k = \frac{\|y\|^2}{N} - \sigma$, can be done as in (5.30) - (5.33). The result of this minimization is the following function, which is to be minimized with respect to σ and X :

$$F = \frac{\alpha^2}{\sigma} + \frac{\beta^2}{(\gamma - \sigma)} \quad (5.69)$$

with $\alpha^2 = b^* B^* \Pi B b$, $\gamma = \frac{\|y\|^2}{N}$ and

$$\beta^2 = \left(\sum_{k=1}^C |\lambda_k| \right)^2 \quad (5.70)$$

where λ_k is the k -th element of the vector $\lambda = (X^* X)^{-1} X^* B b$.

Next, we consider the minimization of (5.69) with respect to $\sigma \in (0, \gamma)$, for fixed X . The corresponding equation for the stationary points of F is :

$$\begin{aligned} F' &= -\frac{\alpha^2}{\sigma^2} + \frac{\beta^2}{(\gamma - \sigma)^2} = 0 \Leftrightarrow \alpha(\gamma - \sigma) = \pm \beta \sigma \Leftrightarrow \\ \sigma &= \frac{\alpha\gamma}{\alpha \pm \beta} \end{aligned} \quad (5.71)$$

Because $\sigma = \frac{\alpha\gamma}{\alpha \pm \beta}$ does not lie in $(0, \gamma)$, the only possible stationary point is :

$$\sigma = \frac{\alpha\gamma}{\alpha + \beta} \quad (5.72)$$

The second-order derivative of F

$$\frac{1}{2} F'' = \frac{\alpha^2}{\sigma^3} + \frac{\beta^2}{(\gamma - \sigma)^3} \quad (5.73)$$

is positive for any $\sigma \in (0, \gamma)$, and therefore (5.72) is a minimum point for F . The corresponding minimum value of F , as a function of X , is given by :

$$\tilde{F} = \frac{(\alpha + \beta)^2}{\gamma} \quad (5.74)$$

The exact minimization of the above function with respect to X does not appear to lead to a simple closed-form solution. However, an approximate solution can be obtained as follows. Under quite general conditions $\beta^2 = \mathcal{O}(1)$ as N increases, whereas $\alpha^2 = \mathcal{O}(N)$ (unless $X = B$). It follows from this observation that, for a reasonably large value of N , the minimization of α in (5.74) is much more important than the minimization of β (note that γ in (5.74) is just a constant). Because $\alpha = 0$ for $X = B$, this means that the true locations of the power peaks are well determined via the minimization of the SPICE criterion with respect to X , which was the fact to be shown.

6. On the SPICE approach for direction of arrival estimation problem

6.1 Introduction and preliminaries

Consider an array processing scenario in which the main problem is to estimate the location parameters of a number of narrowband sources that are present in the array's viewing field. Let Ω denote the set of possible locations, and let θ be a generic location parameter. Also, let $\{\theta_k\}_{k=1}^K$ denote a grid that covers Ω . We assume that the grid is fine enough such that the true location parameters of the existing sources lie on (or, practically, close to) the grid. Under this reasonable assumption we can use the following *non-parametric* model for the output of the array (see, e.g. [98]) :

$$y(t) = \sum_{k=1}^K a_k s_k(t) + \varepsilon(t) \quad t = 1, \dots, M \quad (N \times 1) \quad (6.1)$$

where M is the total number of snapshots, N is the number of sensors in the array, $y(t) \in \mathbb{C}^{N \times 1}$ is the t -th observed snapshot, $a_k \in \mathbb{C}^{N \times 1}$ denotes the array transfer vector (aka manifold or steering vector) corresponding to θ_k , $s_k(t) \in \mathbb{C}$ is the unknown signal impinging on the array from a possible source at θ_k , and $\varepsilon(t) \in \mathbb{C}^{N \times 1}$ is a noise term.

A *sparse* (or semi-parametric) estimation method makes the assumption, reminiscent of the parametric approach, that only a small number of sources exists and therefore that only a few rows of the signal matrix

$$S = \begin{bmatrix} s_1(1) & \cdots & \cdots & s_1(M) \\ \vdots & \vdots & \vdots & \vdots \\ s_K(1) & \cdots & \cdots & s_K(M) \end{bmatrix} \quad (6.2)$$

are different from zero. The estimation problem is then to decide from the data $\{y(t)\}$ which rows of the above matrix are non-zero. Indeed, once this is done, the solution to the location estimation problem, which is usually the main goal of array processing, is immediate : if the row \tilde{k} (let us say) of (6.2) is deemed to be different from zero then we can infer that there is a corresponding source at an estimated location equal to $\theta_{\tilde{k}}$.

The previous formulation of the location problem begs for the use of basic ideas from the area of sparse parameter estimation, or rather simple extensions of those ideas to the present multi-snapshot case. To describe these ideas briefly, let :

$$Y^* = [y(1), \dots, y(M)] \in \mathbb{C}^{N \times M} \quad (6.3)$$

$$S = \begin{bmatrix} s_1^* \\ \vdots \\ s_K^* \end{bmatrix} \in \mathbb{C}^{K \times M} \quad (6.4)$$

$$B^* = [a_1 \cdots a_K] \in \mathbb{C}^{N \times K} \quad (6.5)$$

where the superscript $*$ denotes the conjugate transpose (we denote the matrix in (6.5) by B to reserve the notation A for an extended form of (6.5) that will be introduced in the next section). A direct application of the ℓ_1 -norm minimization principle (see, e.g. [107]) to the present scenario described by (6.1)-(6.5) consists of estimating the matrix S as the solution to the following constrained minimization problem :

$$\min_S \sum_{k=1}^K \|s_k\| \quad \text{s.t. } \|Y^* - B^*S\| \leq \eta \quad (6.6)$$

where $\|\cdot\|$ denotes the Euclidean norm for vectors and the Frobenius norm for matrices, and η is a threshold that must be chosen by the user. Note that the objective in (6.6) is equal to the ℓ_1 -norm of the vector $\{\|s_k\|\}_{k=1}^K$, an observation that shows clearly that this approach is a direct extension of the standard single-snapshot approach of, e.g., [107].

A method for array processing based on (6.6) was pursued in [62]. Note that (6.6) is easily recognized to be an SOCP (second order cone program), see, e.g., [58], which can be efficiently solved provided that N , M and K do not take on too large values. However, in the array processing application this is not necessarily the case : indeed, while $N \approx 10 - 10^2$ (tens to hundreds) is reasonably small, $M \approx 10^2 - 10^3$ and $K \approx 10^2 - 10^6$ (depending on the desired resolution and the dimension of θ (1D, 2D, 3D etc)) can be rather large. For such large dimensional problems the currently available SOCP software (see, e.g., [104]) is too slow to use. In an attempt to overcome this computational problem, [62] suggested a way to reduce the number of columns of Y^* and S in (6.6) to manageable values by means of a singular value decomposition operation.

The approaches that rely on (6.6), such as the one in [62], suffer from a number of problems. First, the motivation of (6.6) is more clearly established in the noise-free case than in the more practical noisy data case. Second, and likely related to the first problem, there exist no clear-cut guidelines for the selection of η in (6.6) (see, e.g., [61] for a critical discussion on this aspect). Finally, solving (6.6) as an SOCP may be too time consuming for some applications in which N , M and especially K take on large values, and the available techniques for reducing the dimensions in (6.6) (such as that suggested in [62]) require the choice of further hyperparameters, besides η , and even so they are only approximate.

In this chapter we present a new method of sparse parameter estimation in models of the type of (6.1). This method, which is called SPICE (**s**pice **i**terative **c**ovariance-based **e**stimation), is obtained using a novel covariance-based fitting approach that was recently introduced in [93] where the focus was on time-series data (the single-snapshot case) as opposed to the array data (the multi-snapshot scenario) considered here. SPICE has a number of useful features that are hardly shared by other sparse estimation methods. In particular, SPICE does not suffer from the problems described in the previous paragraph. Indeed, as will be shown in the next section, SPICE has a sound (covariance-based) statistical motivation which makes it possible to use the method in noisy data scenarios without the need for choosing any hyperparameters. Additionally, the SPICE algorithm has a simple form, and yet it enjoys global convergence properties. We will show that the covariance fitting problem that SPICE solves can be reformulated as an SOCP of the form of (6.6) *but* with $\eta = 0$, with Y^* and S replaced by other matrices with only N columns (typically $M \gg N$), and with row-augmented matrices A and C that, unlike B and S in (6.6), take account of the noise term in the data equation (6.1). This SOCP formulation of the SPICE estimation criterion shows that the ℓ_1 -norm minimization problem in (6.6) can be given a simple statistical motivation based on covariance fitting, provided that the matrices in (6.6) are suitably defined, see the rest of this chapter for details.

6.2 SPICE estimation criterion

Let us assume that

$$E[\varepsilon(t)\varepsilon^*(\bar{t})] = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_N \end{bmatrix} \delta_{t,\bar{t}} \quad (6.7)$$

where E stands for the expectation operator, and

$$\delta_{t,\bar{t}} = \begin{cases} 1 & \text{if } t = \bar{t} \\ 0 & \text{elsewhere.} \end{cases} \quad (6.8)$$

This assumption on the noise term in (6.1) is reasonable in most applications. Let us also assume that the signals $\{s_k(t)\}$ and the noise $\varepsilon(\bar{t})$ are uncorrelated with each other for any (t, \bar{t}) and that :

$$E[s_k(t)s_k^*(\bar{t})] = p_k \delta_{k,\bar{k}} \delta_{t,\bar{t}} \quad (6.9)$$

Then the data snapshots $\{y(1), \dots, y(M)\}$ are uncorrelated with one another and have the following covariance matrix :

$$\begin{aligned}
R &= E[y(t)y^*(t)] = \sum_{k=1}^K p_k a_k a_k^* + \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_N \end{bmatrix} \\
&= [a_1, \dots, a_K I] \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & p_2 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & p_K & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \sigma_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \sigma_N \end{bmatrix} \begin{bmatrix} a_1^* \\ \vdots \\ a_K^* \\ I \end{bmatrix} \quad (6.10)
\end{aligned}$$

$\stackrel{\Delta}{=} A^*PA$

where

$$A^* \stackrel{\Delta}{=} [a_1, \dots, a_K I] \stackrel{\Delta}{=} [a_1, \dots, a_K a_{K+1}, \dots, a_{K+N}] \quad (6.11)$$

$$\begin{aligned}
P &\stackrel{\Delta}{=} \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & p_2 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & p_K & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \sigma_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \sigma_N \end{bmatrix} \\
&\stackrel{\Delta}{=} \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & p_2 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & p_K & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & p_{K+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & p_{K+N} \end{bmatrix} \quad (6.12)
\end{aligned}$$

The assumption in (6.9) that the source signals are spatially uncorrelated, which led to the above expression for the covariance matrix R , does not always hold: in some applications the signals can be correlated or even coherent. However the SPICE method proposed in this chapter is *robust* to this assumption - we refer to [93] (see also [98]) for a theoretical explanation of this robustness property, which will be illustrated numerically in Section 6.5.

We will consider the following covariance fitting criterion for the purpose of parameter estimation (see, e.g., [75] and the references therein):

$$f = \|R^{-1/2}(\hat{R} - R)\hat{R}^{-1/2}\|^2 \quad (6.13)$$

where $R^{-1/2}$ denotes the positive definite square-root of R^{-1} ,

$$\hat{R} = Y^*Y/M \quad (6.14)$$

and the inverses of \hat{R} and R are assumed to exist. Note that R^{-1} exists under weak conditions : indeed, while the "true" values of many $\{p_k\}$ in R may be zero, one typically has $\sigma_k > 0$ ($k = 1, \dots, N$) which renders R nonsingular. Regarding \hat{R} , the inverse of this matrix exists with probability one as long as $M > N$. However, for $M < N$ the sample covariance matrix \hat{R} is singular and therefore (6.13) cannot be used. In the latter case one can estimate the parameters $\{p_k\}$ by minimizing the following criterion, instead of (6.13),

$$\|R^{-1/2}(\hat{R} - R)\|^2 \quad (6.15)$$

The above criterion is the one used in [93] where the focus was on the time series case with $M = 1$. In the present array processing scenario, however, we prefer to consider (6.13) because usually the condition $M > N$ is satisfied and because (6.13) has a statistically stronger motivation than (6.15) : indeed, under certain conditions specified in [75], it can be shown that the parameter estimates minimizing (6.13) are statistically asymptotically (in M) efficient, whereas the estimates obtained from (6.15) are sub-optimal. Nevertheless, we should note that what we do in the following for (6.13) applies with minor modifications to (6.15) as well, see Remark 1 below - this observation can be useful in those array processing applications in which $M < N$ and therefore in which (6.15) should be used in lieu of (6.13).

A simple calculation shows that :

$$\begin{aligned} f &= \text{tr} \left[R^{-1}(\hat{R} - R)\hat{R}^{-1}(\hat{R} - R) \right] \\ &= \text{tr} \left[(R^{-1}\hat{R} - I)(I - \hat{R}^{-1}R) \right] = \text{tr}(R^{-1}\hat{R}) + \text{tr}(\hat{R}^{-1}R) - 2N \end{aligned} \quad (6.16)$$

where

$$\text{tr} \left(\hat{R}^{-1}R \right) = \sum_{k=1}^{K+N} p_k a_k^* \hat{R}^{-1} a_k \quad (6.17)$$

It follows from (6.16) and (6.17) that the minimization of f is equivalent to the minimization of the function:

$$g = \text{tr} \left(\hat{R}^{1/2} R^{-1} \hat{R}^{1/2} \right) + \sum_{k=1}^{K+N} (a_k^* \hat{R}^{-1} a_k) p_k \quad (6.18)$$

Remark 5 A similar calculation shows that in the case of (6.15) the function to be minimized with respect to the unknown powers $\{p_k\}$ is :

$$\text{tr}(\hat{R}R^{-1}\hat{R}) + \sum_{k=1}^{K+N} \|a_k\|^2 p_k \quad (6.19)$$

Owing to the analogy between (6.18) and (6.19), it should come as no surprise that what we do in the following for (6.18) can also be done for (6.19) if necessary (e.g. for $M < N$). ■

The problem of minimizing g in (6.18) with respect to $\{p_k\}$ can be easily shown to be an SDP (semi-definite program), see 7.5, and is therefore *convex* (see, e.g., [16]). Solving (6.18) as an SDP, however, is not recommended due to the fact that the available SDP solvers are too computationally intensive for the values of N , M and K encountered in many array processing applications. Consequently we adopt a different line of attack that consists of replacing the problem of minimizing (6.18) by a related problem, as described next.

It follows from (6.17) that a consistent (in M) estimate of the right-hand side of this equation is given by N . Therefore we can think of reformulating the problem of minimizing g as the following constrained minimization :

$$\min_{\{p_k \geq 0\}} \text{tr}(\hat{R}^{1/2}R^{-1}\hat{R}^{1/2}) \quad \text{s.t. } \sum_{k=1}^{K+N} w_k p_k = 1 \quad (6.20)$$

where

$$w_k = a_k^* \hat{R}^{-1} a_k / N \quad (6.21)$$

Interestingly, the problems (6.18) and (6.20) are not only asymptotically equivalent (as M increases, and under the condition that R in (6.10) can represent the true covariance matrix) but they are *exactly equivalent* (in general) in the sense that their solutions are scaled versions of each other (note that a scaling of $\{p_k\}$ has no effect on source location estimation). This equivalence property is proved in 7.6 where we also show that the problem obtained from (6.18) by constraining the first term to N (a constraint suggested, once again, by asymptotic considerations), instead of constraining the second term as in (6.20), is equivalent to (6.18) and (6.20) as well.

The problem (6.20) is also an SDP, and therefore convex. Furthermore, note that the linear constraint in (6.20) is of the (weighted) ℓ_1 -norm type, and thus it can be expected to be sparsity inducing for the solution to (6.20). Apparently, the criterion in (6.20) was never considered in the previous literature on sparse parameter estimation. On the other hand, interestingly enough, this *type* of criterion occurs frequently in the seemingly unrelated literature on optimal experiment design (see, e.g. [125] and the references of that paper).

6.3 SPICE updating formulas

Let $C \in \mathbb{C}^{(K+N) \times N}$ and consider the following problem :

$$\min_C \text{tr}(C^* P^{-1} C) \quad \text{s.t. } A^* C = \hat{R}^{1/2} \quad (6.22)$$

The solution to (6.22) is given (for fixed P) by

$$C_0 = PAR^{-1}\hat{R}^{1/2} \quad (6.23)$$

and the corresponding minimum value of the function in (6.22) is

$$\begin{aligned} \text{tr}(C_0^* P^{-1} C_0) &= \text{tr}(\hat{R}^{1/2} R^{-1} \hat{R}^{1/2}) \\ &= \text{the original objective in (6.20)} \end{aligned} \quad (6.24)$$

To prove the above assertion, observe that (6.23) follows if we can show that (hereafter the notation $X \geq Y$, with X and Y being Hermitian matrices of appropriate dimensions, means that the difference matrix $X - Y$ is positive semi-definite) :

$$C^* P^{-1} C \geq C_0^* P^{-1} C_0 = \hat{R}^{1/2} R^{-1} \hat{R}^{1/2} \quad \text{s.t. } A^* C = \hat{R}^{1/2} \quad (6.25)$$

By standard properties of partitioned matrices (see, e.g., [88], Lemma A.3) and the fact that $R > 0$, (6.25) holds if and only if the following partitioned matrix is positive semi-definite :

$$\begin{bmatrix} C^* P^{-1} C & \hat{R}^{1/2} \\ \hat{R}^{1/2} & R \end{bmatrix} = \begin{bmatrix} C^* P^{-1} C & C^* A \\ A^* C & A^* PA \end{bmatrix} = \begin{bmatrix} C^* & 0 \\ 0 & A^* \end{bmatrix} \begin{bmatrix} P^{-1} & I \\ I & P \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & A \end{bmatrix} \geq 0 \quad (6.26)$$

The central matrix in (6.26) can be re-written as

$$\begin{bmatrix} P^{-1} & I \\ I & P \end{bmatrix} = \begin{bmatrix} P^{-1/2} \\ P^{1/2} \end{bmatrix} \begin{bmatrix} P^{-1/2} & P^{1/2} \end{bmatrix} \quad (6.27)$$

and thus it is obviously positive semi-definite (because it has the form $X^* X$, with $X = [P^{-1/2} P^{1/2}]$). Therefore (6.25) (and hence (6.23)) is proved and then (6.24) follows via substitution.

To summarize, we have proved above (see (6.24)) that the minimization of the objective in (6.22) with respect to C , for any fixed $P \geq 0$, leads to the original function of P in (6.20). Then it follows that the minimization of (6.22) with respect to C and $\{p_k\}$ yields the same $\{p_k\}$ as the minimization of (6.20). The usefulness of this observation lies in the fact that the minimization of the augmented function in (6.22) can be conveniently done by means of a cyclic algorithm that minimizes (6.22) with respect to C , for fixed P , then minimizes

(6.22) with respect to P , for given C , and so forth until convergence. The solution to the first step of this algorithm was already derived, see (6.23). The solution needed in the second step can also be obtained in closed form. To show how this can be done, let

$$C = \begin{bmatrix} c_1^* \\ \vdots \\ c_{K+N}^* \end{bmatrix} \quad (6.28)$$

and observe that

$$\text{tr}(C^*P^{-1}C) = \text{tr}(P^{-1}CC^*) = \sum_{k=1}^{K+N} \|c_k\|^2/p_k \quad (6.29)$$

6.3.1 The case of different $\{\sigma_k\}$

By the Cauchy-Schwarz inequality

$$\left[\sum_{k=1}^{K+N} w_k^{1/2} \|c_k\| \right]^2 \leq \left[\sum_{k=1}^{K+N} \frac{\|c_k\|^2}{p_k} \right] \left[\sum_{k=1}^{K+N} w_k p_k \right] = \sum_{k=1}^{K+N} \frac{\|c_k\|^2}{p_k} \quad (6.30)$$

From this it follows that the minimization of the objective in (6.22) with respect to $\{p_k\}$ (s.t. $p_k \geq 0$ and $\sum_{k=1}^{K+N} w_k p_k = 1$) gives (for fixed C) :

$$p_k = \frac{\|c_k\|}{w_k^{1/2} \rho}, \quad \rho = \sum_{m=1}^{K+N} w_m^{1/2} \|c_m\| \quad (6.31)$$

and the corresponding minimum value of the objective is

$$\left(\sum_{k=1}^{K+N} w_k^{1/2} \|c_k\| \right)^2 \quad (6.32)$$

Equation (6.31) above provides the solution to the second step of the cyclic algorithm, whereas the solution to the first step is given by (6.23). Combining (6.23) and (6.31) leads to the updating formulas of the *SPICE algorithm* in which only the powers $\{p_k\}$ (that are the quantities of main interest) appear explicitly :

$$p_k^{i+1} = p_k^i \frac{\|a_k^* R^{-1}(i) \hat{R}^{1/2}\|}{w_k^{1/2} \rho(i)} \quad k = 1, \dots, K+N \quad (6.33)$$

$$\rho(i) = \sum_{m=1}^{K+N} w_m^{1/2} p_m^i \|a_m^* R^{-1}(i) \hat{R}^{1/2}\| \quad (6.34)$$

where the index i denotes the iteration number, and $R(i)$ is the matrix R made from $\{p_k^i\}$. The algorithm can be initialized with the power estimates obtained by means of the periodogram method (see, e.g., [98]):

$$p_k^0 = a_k^* \hat{R} a_k / \|a_k\|^4 \quad k = 1, \dots, K+N \quad (6.35)$$

Remark 6 The SPICE algorithm for the alternative covariance fitting criterion in (6.15) (or (6.19)) can be readily derived by paralleling the above calculations. The result is an updating formula similar to (6.33) above with the only difference that $\hat{R}^{1/2}$ in (6.33) and (6.34) should be replaced by \hat{R} and $\{w_k\}$ in (6.21) by $w_k = \|a_k\|^2/\text{tr}(\hat{R})$. ■

6.3.2 The case of identical $\{\sigma_k\}$

In some applications it is known that the noise components in the different elements of the array output vector have the same variance :

$$\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma \quad (6.36)$$

Using this information is important as based on it we can reduce the number of powers that need to be estimated. To derive the necessary modification of SPICE, that takes (6.36) into account, first observe that the minimization of (6.22) with respect to C is not affected by (6.36). However the minimization with respect to $\{p_k\}_{k=1}^K$ and σ , for fixed C , is slightly different. Under the above constraint on $\{\sigma_k\}$, equation (6.29) becomes :

$$\text{tr}(C^*P^{-1}C) = \sum_{k=1}^K \|c_k\|^2/p_k + \sum_{k=K+1}^{K+N} \|c_k\|^2/\sigma \quad (6.37)$$

This function is to be minimized with respect to $\{p_k \geq 0\}$ and $\sigma \geq 0$, subject to

$$\sum_{k=1}^K w_k p_k + \gamma \sigma = 1 \quad (6.38)$$

where

$$\gamma = \sum_{k=K+1}^{K+N} w_k \quad (6.39)$$

By the same argument as above, see (6.30)-(6.32), the solution to this optimization problem is :

$$p_k = \frac{\|c_k\|}{w_k^{1/2} \rho} \quad k = 1, \dots, K \quad (6.40)$$

$$\sigma = \frac{\left[\sum_{k=K+1}^{K+N} \|c_k\|^2 \right]^{1/2}}{\gamma^{1/2} \rho} \quad (6.41)$$

where

$$\rho = \sum_{k=1}^K w_k^{1/2} \|c_k\| + \gamma^{1/2} \left[\sum_{k=K+1}^{K+N} \|c_k\|^2 \right]^{1/2} \quad (6.42)$$

and the corresponding minimum function (for fixed C) is

$$\left(\sum_{k=1}^K w_k^{1/2} \|c_k\| + \gamma^{1/2} \left[\sum_{k=K+1}^{K+N} \|c_k\|^2 \right]^{1/2} \right)^2 \quad (6.43)$$

Inserting the expression (6.23) for C in (6.40) and (6.41) we obtain the following *modified SPICE algorithm* for the case of (6.36):

$$p_k^{i+1} = p_k^i \frac{\|a_k^* R^{-1}(i) \hat{R}^{1/2}\|}{w_k^{1/2} \rho(i)} \quad k = 1, \dots, K \quad (6.44)$$

$$\sigma^{i+1} = \sigma^i \frac{\|R^{-1}(i) \hat{R}^{1/2}\|}{\gamma^{1/2} \rho(i)} \quad (6.45)$$

$$\rho(i) = \sum_{k=1}^K w_k^{1/2} p_k^i \|a_k^* R^{-1}(i) \hat{R}^{1/2}\| + \gamma^{1/2} \sigma^i \|R^{-1}(i) \hat{R}^{1/2}\| \quad (6.46)$$

Initial estimates of the powers, for $k = 1, \dots, K$, can still be obtained with the periodogram (see (6.35)), and σ can be initialized for instance as the average of the N smallest values of $\{p_k^0\}_{k=1}^K$ each multiplied by $\|a_k\|^2$. To motivate this choice for σ^0 note that, at least for M sufficiently large, we have $p_k^0 = a_k^* \hat{R} a_k / \|a_k\|^4 \geq \sigma / \|a_k\|^2$ with the equality being likely to hold for the smallest values of p_k^0 , e.g. for the N smallest such values. This observation implies that (below $\{\tilde{p}_k^0\}_{k=1}^N$ are the N smallest values of the set $\{p_k^0\}$ and $\{\tilde{a}_k\}$ the corresponding manifold vectors)

$$\sigma^0 = \sum_{k=1}^N \tilde{p}_k^0 \|\tilde{a}_k\|^2 / N \quad (6.47)$$

can be expected to be a reasonable estimate of σ .

Because SPICE monotonically decreases the objective function (due to its cyclic operation) and as the minimization problem it solves is convex, we can expect that the algorithm has global convergence properties. That this is indeed the case follows from the general analysis in [125] where it was proved that, under weak conditions (essentially requiring that $p_k^0 > 0$ and that the matrix $R(i)$ stays positive definite as the iteration proceeds), *the limit points of the SPICE iterative process are global solutions to (6.20)*.

6.4 SOCP formulation of SPICE

SPICE takes account of the noise without introducing any hyperparameters to be selected by the user. Indeed, SPICE is a *fully data adaptive* approach that estimates both $\{p_k\}$ and $\{\sigma_k\}$ from the available data snapshots. As already mentioned, to our knowledge, the SPICE criterion was never used in the

previous literature on sparse parameter estimation. It is therefore somewhat unexpected that SPICE can be related to (an extended version of) the classical ℓ_1 -norm minimization approach in (6.6), as we explain in the rest of this section.

As indicated in Section 6.2 the SPICE estimation problem, (6.20), can be cast and solved as an SDP. However it can also be reformulated as an SOCP which is a special case of SDP. It is this reformulation of (6.20) as an SOCP that reveals the connection between SPICE and a certain extended ℓ_1 -norm minimization problem. To reformulate SPICE as an SOCP we proceed in the following way. Instead of minimizing (6.22) cyclically (or alternately) with respect to C and P , we minimize this function with respect to $\{p_k\}$ for arbitrary C . The result of this minimization operation is already available from the calculations in the previous section where it was shown that the minimizing powers are given by (6.31) and the corresponding minimum function by (6.32). It remains to minimize (6.32) with respect to C (under the constraint in (6.22)):

$$\min_C \sum_{k=1}^{K+N} w_k^{1/2} \|c_k\| \quad \text{s.t. } A^*C = \hat{R}^{1/2} \quad (6.48)$$

Once C is obtained from (6.48), $\{p_k\}$ can be calculated using (6.31).

The above problem, which is an SOCP (see 7.5), is similar to the ℓ_1 -norm minimization problem (6.6). However there are also significant differences between these two problems, as well as between the ways in which they have been obtained :

- i) The ℓ_1 -norm in (6.48) has a weighted form, with the weights determined by the data. It is well known that $w_k = a_k^* \hat{R}^{-1} a_k / N$ is an estimate of the inverse power corresponding to the k -th point of the location grid (see, e.g., [98]). Therefore, the smaller the said power the larger the $w_k^{1/2}$ and consequently more weight is put on minimizing the associated $\|c_k\|$ in (6.48), which intuitively is the way it should be. Furthermore, the ℓ_1 -norm objective in (6.48) follows naturally from the statistically sound covariance fitting criterion in (6.13), in contrast with the similar objective in (6.6) whose motivation is essentially heuristical.
- ii) The matrices A and C in (6.48) comprise extra rows that account for the noise in the data, unlike the similar quantities B and S in (6.6).
- iii) The inequality constraint in (6.6) is replaced in (6.48) by a hyperparameter-free equality constraint where $\hat{R}^{1/2}$ is used in lieu of Y^* in (6.6). Note that the matrix C in (6.48) has N columns instead of M columns as S in (6.6), which evidently makes (6.48) more convenient computationally (typically $M \gg N$). This dimensional reduction also follows naturally from the employed covariance fitting criterion, whereas the idea suggested in [62] for reducing the number of columns of Y^* and S in (6.6) was motivated on more or less heuristical grounds and its use depends on the selection of an additional hyperparameter, besides η .

The reader might wonder what is the relationship between C in (6.48) and S in (6.6). To clarify this aspect, observe that we can write the data equation (6.1) in the following matrix form :

$$Y^* = B^*S + \Delta \quad (6.49)$$

where

$$\Delta = [\varepsilon(1), \dots, \varepsilon(M)] \quad (6.50)$$

It follows from (6.49) that we can assume, as in (6.6), that $\|Y^* - B^*S\| \leq \eta$ provided that η is "sufficiently large" ; furthermore, a proper selection of η would require information about the noise variance(s) and the noise distribution, which is usually unavailable, and even then the inequality constraint in (6.6) could not in general be made to hold surely for any finite practically convenient value of η . On the other hand, it also follows from (6.49) that

$$Y^* = A^* \begin{bmatrix} S \\ \Delta \end{bmatrix} \quad (6.51)$$

and therefore that

$$A^* \underbrace{\begin{bmatrix} S \\ \Delta \end{bmatrix}}_{\stackrel{\Delta}{=} C} Y \hat{R}^{-1/2} / M = \hat{R}^{1/2} \quad (6.52)$$

The above equation clarifies the relationship between C and S . Unlike (6.6), the equality constraint (6.52) holds deterministically and it does not involve any hyperparameter. Note also that if S is row-sparse then so is the matrix $SY\hat{R}^{-1/2}/M$ in (6.52). Regarding the other matrix appearing in (6.52), viz. $\Delta Y \hat{R}^{-1/2}/M$, it tends to zero as the signal-to-noise ratio (SNR) increases ; however, for small or medium SNR values this matrix is not row-sparse, which may be perceived as a downside of (6.52), although a relatively minor one as the matrix C can still be considered to be row-sparse because $K \gg N$.

Whenever *the constraint $\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma$ is enforced*, the above reformulation of the SPICE estimation problem should be changed as follows. First, C is obtained as the solution of the following SOCP (see (6.43)) :

$$\min_C \sum_{k=1}^K w_k^{1/2} \|c_k\| + \gamma^{1/2} \left[\sum_{k=K+1}^{K+N} \|c_k\|^2 \right]^{1/2} \quad \text{s.t. } A^*C = \hat{R}^{1/2} \quad (6.53)$$

(see 7.5 for the formulation of the above problem as an SOCP). Then $\{p_k\}$ and σ are determined using (6.40)-(6.42).

According to our albeit limited experience, the SPICE multiplicative algorithm in (6.33)-(6.34) (or (6.44) -(6.46)) is usually visibly faster than the SOCP-based algorithm that was outlined in this section. However, the SOCP formulation of the SPICE estimation problem remains important for understanding the connection between the proposed approach and the standard one based on ℓ_1 -norm minimization.

6.5 Numerical illustrations and concluding remarks

In this section we illustrate numerically the performance of the proposed methods and compare it with the performance of some standard methods from the literature. We will consider both a uniform linear array (ULA) and a non-uniform linear array (NULA). In the ULA case the sensors are uniformly placed with a spacing of $\frac{\lambda}{2}$, where λ denotes the wavelength of the sources. The inter-element spacings in the case of NULA are as shown in Fig 6.1. The number of sensors in the array is $N = 10$ for ULA and $N = 100$ for NULA. In the ULA case the number of snapshots is $M = 200$ whereas it is $M = 100$ in the NULA case. The steering vector a_k for the ULA, corresponding to a direction of arrival (DOA) equal to θ_k , is given by:

$$a_k = \begin{bmatrix} e^{i\pi \sin(\theta_k)} \\ \vdots \\ e^{i\pi N \sin(\theta_k)} \end{bmatrix} \quad (6.54)$$

(observe that $\|a_k\|^2 = N$ is a constant). The steering vector for the NULA can be similarly defined. The interval for the DOA is $\Omega = (-90^\circ, 90^\circ]$. We use a uniform grid $\{\theta_k\}_{k=1}^K$ to cover Ω , with a step of 0.1° , which means that $K = 1800$.

The source signals $\{s_k(t)\}$, see (6.1), have constant modulus, which is usually the situation in communications applications. We will consider cases with both uncorrelated and coherent sources. The noise term in (6.1) is chosen to be white, both temporally and spatially, and Gaussian distributed with zero mean and variance σ .

6.5.1 Fixed sources : DOA estimation

In this subsection we consider DOA estimation of fixed sources using the ULA. The data samples were simulated using equation (6.1) with three sources at $\theta_1 = 10^\circ$, $\theta_2 = 40^\circ$, $\theta_3 = 55^\circ$ and the following signals $s_1(t) = 3e^{i\varphi_1(t)}$, $s_2(t) = 10e^{i\varphi_2(t)}$, and $s_3(t) = 10e^{i\varphi_3(t)}$, where the phases $\{\varphi_k(t)\}_{k=1}^3$ were independently and uniformly distributed in $[0, 2\pi]$. In the simulation with coherent sources, the sources at θ_1 and θ_3 were the coherent ones (i.e. they had identical phases), whereas the source at θ_2 was uncorrelated to them. The noise variance was varied to control the signal-to-noise ratio defined as :

$$\text{SNR} = 10 \log\left(\frac{100}{\sigma}\right) = 20 - 10 \log \sigma \quad [\text{dB}]$$

The following methods were used for DOA estimation :

- M_1 : The periodogram (PER), see (6.35).
- M_2 : The iterative adaptive approach (IAA), see [122].
- M_3 : The multiple signal classification method (MUSIC), see e.g. [98].
- M_4 : SPICE, see (6.15), (6.19) as well as Remark 2 and [93].

M_5 : The enhanced SPICE or, for short, SPICE-plus (SPICE+), see (6.44)-(6.46).

Note that IAA was shown in [122] to outperform the method of [62] and its variations in the literature, which is why we selected IAA for the present comparison. As a cross-check and for comparison's sake we have also computed the SPICE+ estimates by means of an SDP solver (see [104] for the latter). The so-obtained estimates were identical (within numerical accuracy) to those provided by the SPICE+ algorithm; however, as expected, the SDP solver was either much slower or, worse, it could not be executed for instance due to memory problems.

In Fig 2 we show the average root mean-square errors (RMSEs) of the DOA estimates obtained with $M_1 - M_5$ in 1000 Monte-Carlo runs, for several SNR values :

$$\text{RMSE} = \left[\frac{1}{3000} \sum_{k=1}^3 \sum_{m=1}^{1000} (\hat{\theta}_k^m - \theta_k)^2 \right]^{1/2}$$

where $\hat{\theta}_k^m$ denotes the estimate of θ_k in the m -th Monte-Carlo run. In this figure, the top plot shows the RMSEs for uncorrelated sources and the bottom plot shows the RMSEs for coherent sources. As can be seen from this figure, the PER estimates of the DOA's have poor accuracy in all cases under study, due to a significant bias (which does not decrease as the SNR increases). Regarding IAA, this method provides competitive estimates only for $\text{SNR} \geq 0\text{dB}$. The parametric method of MUSIC, which requires knowledge on the number of sources in the data, yields reasonably accurate DOA estimates in the uncorrelated source case (at least for $\text{SNR} \geq 0\text{dB}$) but completely fails in the coherent source case (as expected). The proposed methods of SPICE and SPICE+ give the best performance in the cases considered : in particular their threshold SNR appears to be lower by some 10dB than the threshold SNR of IAA and of MUSIC. For uncorrelated sources, SPICE+ is more accurate than SPICE, whereas SPICE outperforms SPICE+ in the coherent source case. From a computational standpoint SPICE+ converged faster than SPICE presumably due to the data-dependent weights used by SPICE+.

6.5.2 Mobile sources : DOA tracking

In this subsection, we consider DOA estimation of mobile sources using the NULA. The data samples were generated assuming two mobile sources, one moving linearly from 30° to 60° and the other from 60° to 30° , in steps of 0.03° , over a course of 1000 data snapshots. The uncorrelated signals of these two sources were given by $s_1(t) = 10e^{i\varphi_1(t)}$ and $s_2(t) = 10e^{i\varphi_2(t)}$, where the phases $\{\varphi_k(t)\}$ were independently and uniformly distributed in $[0, 2\pi]$.

In this example, only the methods M_1 , M_4 and M_5 are considered. At any given time instant t , the sample covariance matrix \hat{R} is formed from the most recent 100 data snapshots and the DOA estimates are re-computed by each

of the three methods. As before, the SPICE methods, M_4 and M_5 , are initialized with the PER estimate. Alternatively, one may think of initializing the SPICE methods with their respective estimates obtained from the previous data window; however, as discussed in Section 6.3.2, for global convergence SPICE should be initialized with a dense estimate rather than a sparse one. The SPICE methods were iterated only 5 times except at $t = 100$ where they were applied for the first time and were iterated till convergence. We have also tried a larger number of iterations, such as 20, but noticed no significant improvement in the DOA estimates (compare Figs 6.3(c) and (d) below). The SNR in this simulation was 20 dB (however very similar results were obtained for SNR = 10 dB and 0 dB).

Fig 6.3 shows the plots of DOA estimates vs. t for the three considered methods. In each plot, the estimates shown at any time t were obtained as the locations of the two largest peaks in the spatial spectrum provided by the method in question. As can be seen from the figure, PER performs poorly as it cannot track either the source moving from 30° to 45° or the one moving from 45° to 30° . Note that at each time instant, the data window (made of the most recent 100 data snapshots) comprises two sets of DOAs each with a width of 3° . For instance, at $t = 400$, the data window will contain signals with DOAs lying in the interval $[39^\circ, 42^\circ]$ and, respectively, in $[48^\circ, 51^\circ]$. PER often picks wrongly the two peaks only from the set with larger DOAs. On the other hand, the SPICE methods correctly pick the two peaks from both sets which leads to the band-like appearance of the corresponding plots in Figs 6.3(b)-(d); observe that, as expected, the width of the bands in these figures (measured along the DOA axis) is equal to 3° . Note also that the true DOA trajectories are well approximated by the upper edge of the bands in Figs 6.3(b)-(d) corresponding to the increasing DOA and by the lower edge of the bands associated with the decreasing DOA - this behavior, which was expected in view of the above discussion, is in fact quite stable with respect to the SNR (in simulations not shown here, we have observed that decreasing the SNR from 20 dB to 0 dB caused only 1 DOA estimate, out of 2000, to lie outside bands similar to those in Figs 6.3(b)-(d)). To provide further insight into this type of behavior, a single source moving along a sinusoidal trajectory was considered. Fig 6.4 shows the DOA estimates for this source obtained with SPICE+. It is clear from this plot that the width of the band decreases till $t = 500$, where it is nearly zero, and then starts increasing again; hence, as expected, the width of the band is proportional to the slope of the DOA variation. Furthermore in this case, too, the true DOA trajectory is well approximated by the upper edge of the band (for increasing DOA) and the lower edge (for decreasing DOA).

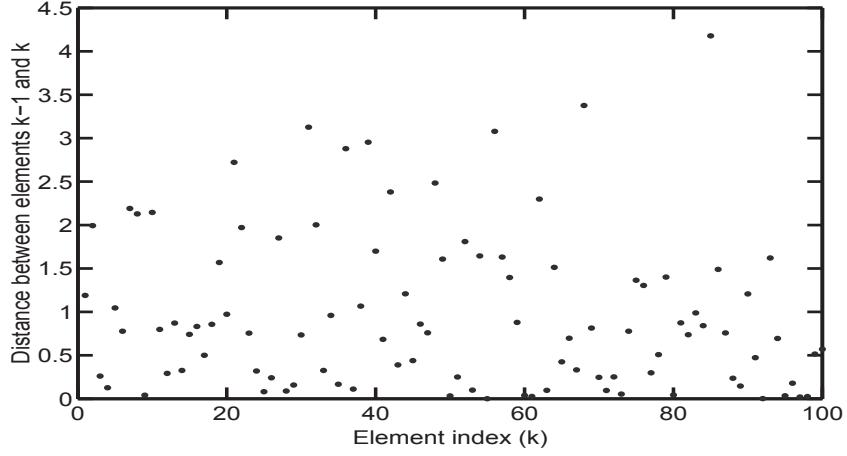


Figure 6.1. Inter-element spacings (in units of $\lambda/2$) for the NULA with $N = 100$.

6.6 Some SDP and SOCP formulations

First we show that the problem of minimizing (6.18), s.t. $p_k \geq 0$ ($k = 1, \dots, K+N$), can be formulated as an SDP. The proof that the same is true for (6.20) is similar and therefore its details are omitted. Let

$$\hat{R}^{1/2} = \begin{bmatrix} r_1^* \\ \vdots \\ r_N^* \end{bmatrix} \quad (6.55)$$

and let

$$v_k = \left(a_k^* \hat{R}^{-1} a_k \right) \quad (6.56)$$

Using this notation we can re-write the function g in (6.18) as :

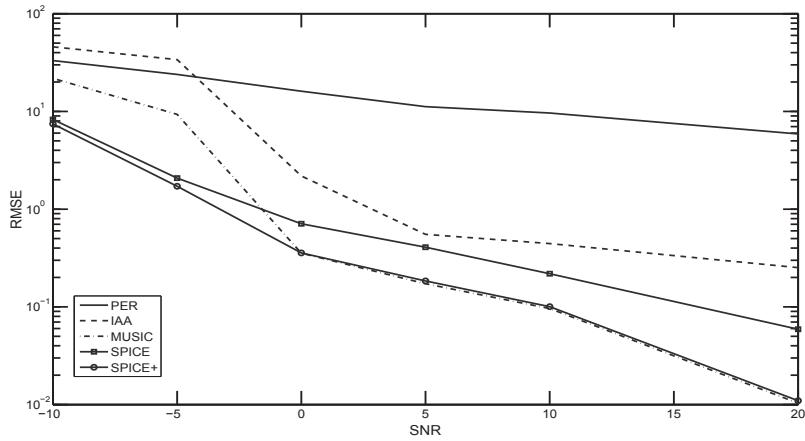
$$g = \sum_{k=1}^N r_k^* R^{-1} r_k + \sum_{k=1}^{K+N} v_k p_k \quad (6.57)$$

Let $\{\alpha_k\}_{k=1}^N$ be auxiliary variables satisfying $\alpha_k \geq r_k^* R^{-1} r_k$, or equivalently

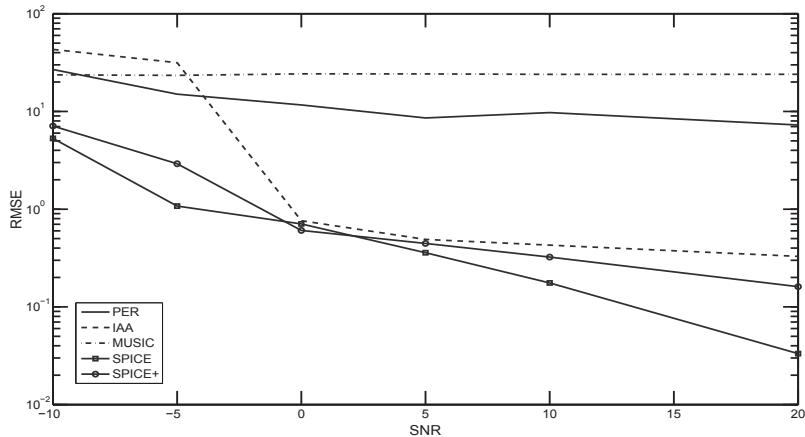
$$\begin{bmatrix} \alpha_k & r_k^* \\ r_k & R \end{bmatrix} \geq 0 \quad (6.58)$$

Then the minimization problem under discussion can be stated as :

$$\begin{aligned} \min_{\{\alpha_k\}, \{p_k\}} \quad & \sum_{k=1}^N \alpha_k + \sum_{k=1}^{K+N} v_k p_k \quad \text{s.t. } p_k \geq 0 \quad k = 1, \dots, K+N \\ & \begin{bmatrix} \alpha_k & r_k^* \\ r_k & R \end{bmatrix} \geq 0 \quad k = 1, \dots, N \end{aligned} \quad (6.59)$$

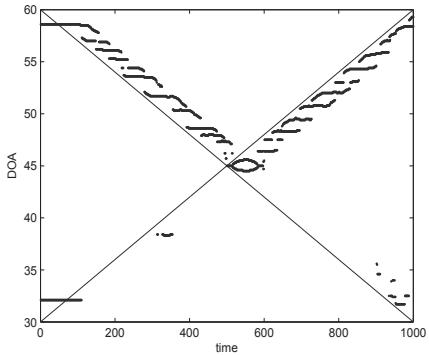


(a) Uncorrelated sources

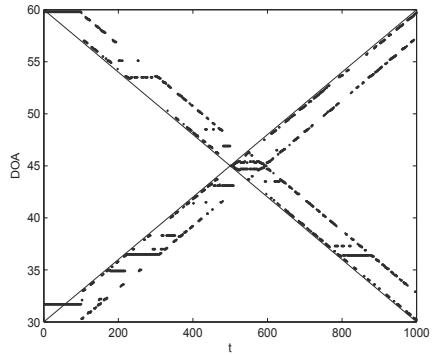


(b) Coherent sources

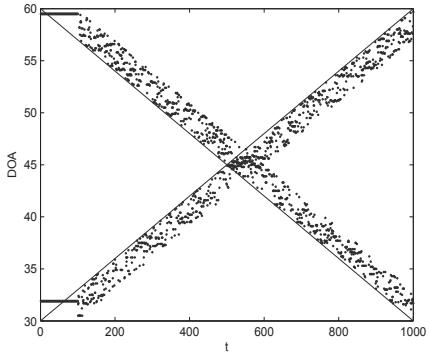
Figure 6.2. Estimation performance of PER, IAA, MUSIC, SPICE and SPICE+ (the ULA case) : RMSE vs SNR for the DOA estimates obtained in 1000 Monte-Carlo runs.



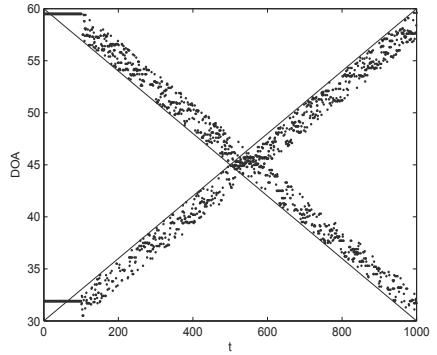
(a) PER



(b) SPICE with 5 iterations



(c) SPICE+ with 5 iterations



(d) SPICE+ with 20 iterations

Figure 6.3. Tracking performance of PER, SPICE and SPICE+ (the NULA case). The solid lines denote the trajectories of the true DOAs.

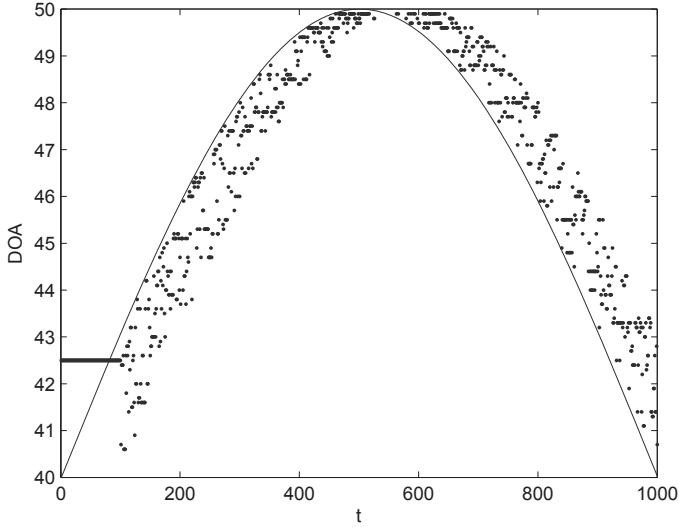


Figure 6.4. Tracking performance of SPICE+ for a sinusoidal trajectory (the NULA case). The solid line denotes the true DOA trajectory.

which is an SDP (as R is a linear function of $\{p_k\}$) [58] [16].

Next, we show that (6.48) can be cast as an SOCP. Let $\{\beta_k\}$ denote $(K+N)$ auxiliary variables that satisfy :

$$\beta_k \geq w_k^{1/2} \|c_k\| \quad k = 1, \dots, K+N \quad (6.60)$$

Then simply observe that (6.48) can be re-written as :

$$\begin{aligned} \min_{\{\beta_k\}, C} & \sum_{k=1}^{K+N} \beta_k \quad \text{s.t.} \quad \|c_k\| \leq \beta_k / w_k^{1/2} \quad k = 1, \dots, K+N \\ & A^* C = \hat{R}^{1/2} \end{aligned} \quad (6.61)$$

which is an SOCP [58].

Finally, we show how to re-formulate (6.53) as an SOCP. Let $\{\beta_k\}$ still denote some auxiliary variables (but now only $K+1$ of them) and constrain them to satisfy :

$$\begin{aligned} \beta_k & \geq w_k^{1/2} \|c_k\| \quad k = 1, \dots, K \\ \beta_{K+1} & \geq \gamma^{1/2} \left[\sum_{k=1}^{K+N} \|c_k\|^2 \right]^{1/2} \end{aligned} \quad (6.62)$$

Using $\{\beta_k\}$ we can re-state (6.53) in the following form :

$$\begin{aligned} \min_{\{\beta_k\}, C} \sum_{k=1}^{K+1} \beta_k & \quad \text{s.t.} \quad \|c_k\| \leq \beta_k / w_k^{1/2} \quad k = 1, \dots, K \\ & \left\| \begin{bmatrix} c_{K+1} \\ \vdots \\ c_{K+N} \end{bmatrix} \right\| \leq \beta_{K+1} / \gamma^{1/2} \\ A^* C & = \hat{R}^{1/2} \end{aligned} \quad (6.63)$$

which, once again, is an SOCP.

6.7 Equivalence of (6.18) and (6.20)

Let P_0 and P_1 denote the minimization problems corresponding to (6.18) and (6.20). Therefore :

$$P_0 : \min_{\{p_k \geq 0\}} \text{tr}(R^{-1}\hat{R}) + \text{tr}(\hat{R}^{-1}R) \quad (6.64)$$

and

$$P_1 : \min_{\{p_k \geq 0\}} \text{tr}(R^{-1}\hat{R}) \quad \text{s.t.} \quad \text{tr}(\hat{R}^{-1}R) = 1 \quad (6.65)$$

Note that we have replaced N by 1 in the right-hand side of the constraint in (6.65) for notational convenience (this replacement has only a scaling effect on the solution of P_1). For completeness, we will also consider the following problem obtained by constraining the first term in (6.64) to one :

$$P_2 : \min_{\{p_k \geq 0\}} \text{tr}(\hat{R}^{-1}R) \quad \text{s.t.} \quad \text{tr}(R^{-1}\hat{R}) = 1 \quad (6.66)$$

First we prove the equivalence of P_1 and P_2 . By making use of an auxiliary variable α we can re-write (6.65) and (6.66) as follows :

$$P_1 : \min_{\alpha, \{p_k \geq 0\}} \alpha \quad \text{s.t.} \quad \text{tr}(R^{-1}\hat{R}) = \alpha \quad \text{and} \quad \text{tr}(\hat{R}^{-1}R) = 1 \quad (6.67)$$

$$P_2 : \min_{\alpha, \{p_k \geq 0\}} \alpha \quad \text{s.t.} \quad \text{tr}(\hat{R}^{-1}R) = \alpha \quad \text{and} \quad \text{tr}(R^{-1}\hat{R}) = 1 \quad (6.68)$$

Let $\{\tilde{p}_k = \alpha p_k\}$ (note that $\alpha > 0$) and reformulate P_1 as :

$$P_1 : \min_{\alpha, \{\tilde{p}_k \geq 0\}} \alpha \quad \text{s.t.} \quad \text{tr}(\hat{R}^{-1}\tilde{R}) = \alpha \quad \text{and} \quad \text{tr}(\tilde{R}^{-1}\hat{R}) = 1 \quad (6.69)$$

where \tilde{R} is made from $\{\tilde{p}_k\}$. It follows from (6.68) and (6.69) that if $\{p_k\}$ is the solution to P_2 then $\{p_k/\alpha\}$ is the solution to P_1 , and thus the proof of the equivalence between P_1 and P_2 is concluded.

Next, we consider P_0 and P_2 . By writing down the Karush-Kuhn-Tucker conditions for P_0 it can be shown that the two terms of the objective in (6.64) must be equal to one another at the minimizing $\{p_k\}$. A more direct proof of this fact runs as follows. Assume that $\{p_k\}$ is the optimal solution of P_0 and that at $\{p_k\}$ we have :

$$\text{tr}(R^{-1}\hat{R}) = \rho^2 \text{tr}(\hat{R}^{-1}R) \quad (6.70)$$

for some $\rho^2 \neq 1$; note that the scaling factor in (6.70) must necessarily be positive, which is why we wrote it as ρ^2 ; below we will let $\rho > 0$ denote the square root of ρ^2 . We will show in the following that the previous assumption leads to a contradiction : $\{p_k\}$ cannot be the solution to P_0 if $\rho \neq 1$. To do so let $\{\tilde{p}_k = \rho p_k\}$ and observe that, at $\{\tilde{p}_k\}$, the two terms in the objective of P_0 are identical :

$$\text{tr}(\tilde{R}^{-1}\hat{R}) = \text{tr}(\hat{R}^{-1}\tilde{R}) \quad (6.71)$$

Making use of (6.70) and (6.71) along with the assumption that $\{p_k\}$ is the solution to P_0 , we obtain the following inequality :

$$\text{tr}(\hat{R}^{-1}R)(1 + \rho^2) < 2\text{tr}(\hat{R}^{-1}\tilde{R}) = 2\rho \text{tr}(\hat{R}^{-1}R) \quad (6.72)$$

or, equivalently,

$$(1 - \rho)^2 < 0 \quad (6.73)$$

which cannot hold, and hence the solution to P_0 must satisfy (6.70) with $\rho = 1$. Using this fact we can re-formulate P_0 as follows :

$$P_0 : \min_{\alpha, \{p_k \geq 0\}} \alpha \text{ s.t. } \text{tr}(R^{-1}\hat{R}) = \alpha \text{ and } \text{tr}(\hat{R}^{-1}R) = \alpha \quad (6.74)$$

or, equivalently (as $\alpha > 0$),

$$P_0 : \min_{\beta, \{\tilde{p}_k \geq 0\}} \beta \text{ s.t. } \text{tr}(\hat{R}^{-1}\tilde{R}) = \beta \text{ and } \text{tr}(\tilde{R}^{-1}\hat{R}) = 1 \quad (6.75)$$

where $\{\tilde{p}_k = \alpha p_k\}$, as before, and $\beta = \alpha^2$ is a new auxiliary variable. Comparing (6.75) and (6.68) concludes the proof of the equivalence of P_0 and P_2 .

7. On two user parameter free sparse parameter estimation approaches : SPICE and LIKES

7.1 Introduction and problem formulation

Consider the following linear model:

$$\begin{aligned} y &= \sum_{k=1}^M a_k x_k + e \\ &= [a_1, \dots, a_M I] \begin{bmatrix} x \\ e \end{bmatrix} = B\beta \end{aligned} \quad (7.1)$$

where

$$\begin{aligned} x &= [x_1, \dots, x_M]^T \\ \beta &= [x^T \ e^T]^T \\ B &= [a_1, \dots, a_M I] \triangleq [b_1, \dots, b_{M+N}]. \end{aligned} \quad (7.2)$$

Furthermore, in (7.1) $y \in \mathbb{C}^N$ denotes the observation vector, $\{a_k \in \mathbb{C}^N\}_{k=1}^M$ is a set of given vectors, $\{x_k \in \mathbb{C}\}_{k=1}^M$ are unknown parameters, and $e \in \mathbb{C}^N$ is a noise term; the matrix $B \in \mathbb{C}^{N \times (M+N)}$ and the vector $\beta \in \mathbb{C}^{M+N}$, introduced in (7.2), are for later use. A number of both linear *and* nonlinear estimation problems occurring in biostatistics, temporal and spatial spectral analysis, radar imaging, astronomy, magnetic resonance imaging and so on (see, e.g., [93] [34] [60] [61] [124] [123] [30] and the many references there) can be reduced to the estimation of x in the above linear model with $M \gg N$ (i.e. scarce data) and with only a few elements of x different from zero (i.e., a sparse parameter vector). Note that if the noise vector e in (7.1) were assumed to be sparse then the parameter vector x could be exactly recovered from y , under fairly weak conditions, for instance by the methods proposed later on in the paper (see, e.g., [120]). However, we do *not* make this assumption here and therefore our problem is the estimation, rather than the exact recovery, of x .

There are several methods in the literature that can be used to estimate the sparse parameter vector in (7.1). However, most of these methods require the selection of one or more user parameters (also called hyperparameters), which is usually a daunting task (see, e.g., the lucid discussion in [61]). In [93] [34] we have recently introduced a **SParse Iterative Covariance-based Estimation** (SPICE) method that does not suffer from this drawback: SPICE, which is derived from a statistically and computationally sound covariance

fitting criterion, does not require the choice of any hyperparameters. We revisit here the derivation of SPICE with the purpose of streamlining it and also of providing further insights into this method. In particular, the focus in [93] and [34] was on temporal and, respectively, spatial spectral analysis, and owing to this focusing the estimation of x in (7.1) was less stressed than that of other (related) parameters, see Section 7.2 for details. Here we change the emphasis and treat x as the parameter vector of primary interest, which is the case in several applications.

The principle of maximum likelihood is generally considered to be statistically more sound than covariance (or moment) fitting. We make use of this principle, in a manner similar to [109] [119] [117], to derive a novel method for estimating x in (7.1), which we designate by the acronym LIKES (**L**I**K**e*l*elihood-based **E**stimation of **S**p*a*re parameters). The concise derivation of LIKES, presented in Section 7.3, relies on that of SPICE in Section 7.2; indeed, we exploit the links between LIKES and SPICE to simplify the description of the steps of LIKES.

In Section 7.4 we present numerical evidence that lends support to the fact that LIKES can be expected to be more accurate than SPICE at the cost of an increased computational burden. The numerical examples also show that SPICE and LIKES provide more accurate parameter estimates than two competitive algorithms, viz. Iterative Reweighted ℓ_1 -norm minimization (IRL1) [24] [128] and Basis Pursuit (BP) [25].

7.2 SPICE

We will make the working assumption that the elements of β are random variables that are uncorrelated to each other and which have zero means and variances denoted by $\{p_k\}_{k=1}^M$ for $\{x_k\}_{k=1}^M$ and $\{\sigma_k\}_{k=1}^N$ for $\{e_k\}_{k=1}^N$. Under this assumption the covariance matrix of y is given by:

$$R = E(yy^*) = BPB^* \quad (7.3)$$

where the superscript * denotes the conjugate transpose, and

$$P = \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & p_M & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \sigma_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \sigma_N \end{bmatrix} \quad (7.4)$$

$$\stackrel{\Delta}{=} \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & p_M & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & p_{M+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & p_{M+N} \end{bmatrix}.$$

The SPICE estimation metric is the following weighted covariance fitting criterion (see [93] [34] [75] and also the references therein) :

$$\|R^{-1/2}(R - yy^*)\|^2 \quad (7.5)$$

where $\|\cdot\|$ denotes the Frobenius norm for matrices (as well as the Euclidean norm for vectors), and $R^{-1/2}$ is a Hermitian square-root of the inverse matrix R^{-1} (which is assumed to exist). The following comments on (7.3) and (7.5) are in order:

- The type of model (7.3) for the data covariance matrix has been sometimes considered in the literature. Most commonly this model was used due to its convenience rather than its veracity. However, the estimation methods based on it are known to be robust to mismodeling (see, e.g., [93] [34] [98]).
- In some cases it may be known that the noise elements have the same variance : $\sigma_1 = \cdots = \sigma_N$. Both SPICE and LIKES (see Section 7.3) can be readily modified to take this information into account : see [93] [34] for SPICE ; the modification of LIKES is similar. However, for the sake of conciseness, we will omit any discussion on imposing this condition on $\{\sigma_k\}$ and refer instead to the cited references for details on this aspect.
- We make a similar remark on the case of multiple data vectors (also known as snapshots), in which the vectors y and x in (7.1) should be

replaced by matrices Y and X with X being row-wise sparse. We refer the interested reader to [34] for details on the extension of SPICE to this case; the extension of LIKES is similar.

- We will estimate the power vector

$$p = [p_1, \dots, p_{M+N}]^T \quad (p_k \geq 0) \quad (7.6)$$

by minimizing the covariance fitting criterion in (7.5). In some cases, such as in the spectral analysis applications considered in [93] [34], estimating p may be deemed to be sufficient. Indeed estimates of $\{p_k\}$ can be enough to determine whether $\{|x_k|\}$ are “large” or “small”, which is what is mainly required for signal detection in the said cases. However, $\{p_k\}$ do not contain any information on the phases of $\{x_k\}$, which are also of interest in some applications. Furthermore, the direct calculation even of $\{|x_k|\}$ from $\{p_k\}$ may not be always possible due to some unknown scaling factor that is involved (see, e.g., [93]). With these facts in mind, here we put the emphasis on estimating x as the parameter vector of main interest, somewhat in contrast to what has been done in [93] [34]. While we still consider the minimization of (7.5) with respect to (wrt) $\{p_k\}$, we show that an estimate of β , and hence of $\{x_k\}$ in particular, occurs naturally in the process of solving this minimization problem. As might have been expected, this estimate of the realization of β that led to the observed data vector y has the following maximum a posteriori (MAP)-like form (e.g., [119]):

$$\beta = PB^*R^{-1}y \quad (7.7)$$

see below for details.

Returning to (7.5), a simple calculation shows that this fitting criterion can be re-written as:

$$\begin{aligned} & \text{tr}[(I - yy^*R^{-1})(R - yy^*)] \\ &= \text{tr}(R) + \|y\|^2y^*R^{-1}y - 2\|y\|^2 \end{aligned} \quad (7.8)$$

where

$$\text{tr}(R) = \sum_{k=1}^{M+N} p_k \|b_k\|^2. \quad (7.9)$$

It follows from (7.8) and (7.9) that the minimization problem of interest here is the following:

$$\min_p y^*R^{-1}y + \sum_{k=1}^{M+N} w_k^2 p_k \quad (7.10)$$

where the weights

$$w_k = \frac{\|b_k\|}{\|y\|} \quad (7.11)$$

do not depend on p . At this point we remark on the fact that in [93] [34] the second term in (7.10) was constrained to be equal to one : $\sum_{k=1}^{M+N} w_k^2 p_k = 1$.

However, we will not impose this constraint here, but instead will consider (7.10) as it stands. The benefit of doing so is twofold : i) the SPICE algorithm obtained from (7.10) is slightly simpler than the version in [93]; and ii) the $\{p_k\}$ obtained from (7.10) are related to $\{|x_k|\}$ via a known scaling factor (see (7.22) below) and thus, if desired, $\{|x_k|\}$ can be expediently calculated from $\{p_k\}$; note that for the $\{p_k\}$ obtained with the SPICE version in [93] [34], the said scaling factor depends on unknown quantities.

The minimization problem in (7.10) is *convex*. Indeed, it can be cast as the following semi-definite program (SDP) [16] (with α being an auxiliary variable):

$$\begin{aligned} & \min_{p, \alpha} \alpha + \sum_{k=1}^{M+N} w_k^2 p_k \\ & \text{s.t. } \begin{bmatrix} \alpha & y^* \\ y & R \end{bmatrix} \geq 0. \end{aligned} \quad (7.12)$$

However we do not recommend obtaining the solution to (7.10) by solving the above SDP. The reason is that the currently available SDP solvers for (7.12) are too time consuming for the values of N and M encountered in many applications (as an example, using a state-of-the art SDP solver for (7.12) with $N = 50$ and $M = 10^3$, which is a medium-size case, takes about one hour on a reasonably powerful PC).

Interestingly enough, (7.10) can also be cast as a second-order cone program (SOCP), which can be solved much more efficiently than the SDP in (7.12) (in less than five minutes for the example in the previous paragraph). This fact follows essentially from the analysis of the multi-snapshot case in [34], but it was not shown explicitly for (7.10). In the next sub-section, we provide a simple proof of the SOCP reformulation of (7.10).

7.2.1 SOCP-based solver

Consider the following augmented problem:

$$\begin{aligned} & \min_{p, \beta} \beta^* P^{-1} \beta + \sum_{k=1}^{M+N} w_k^2 p_k \\ & \text{s.t. } B\beta = y. \end{aligned} \quad (7.13)$$

The use of the symbol β to denote the extra variables in (7.13) is not accidental : indeed, the constraint in (7.13) is nothing but the data equation (7.1). Similarly to (7.10), the problem (7.13) can also be shown to be *convex*. Note that if some $\{p_k\}$ are equal to zero, then P^{-1} in (7.13) should be replaced by the pseudo-inverse of P ; the following analysis carries over the latter case with only relatively minor modifications.

The property of (7.13), which is important here, is that the minimization wrt β yields the original problem. Specifically :

$$\min_{\beta} \beta^* P^{-1} \beta = y^* R^{-1} y \quad \text{s.t. } B\beta = y. \quad (7.14)$$

Consequently, the solutions p to (7.10) and (7.13) are identical. To prove (7.14), we need to show that

$$\beta^* P^{-1} \beta \geq y^* R^{-1} y \quad \text{s.t. } B\beta = y \quad (7.15)$$

or equivalently (making use of the constraint) :

$$\beta^* P^{-1} \beta \geq \beta^* B^* R^{-1} B \beta \quad (7.16)$$

which holds if and only if

$$P^{-1} \geq B^* R^{-1} B \quad (7.17)$$

(for two Hermitian matrices A and B , the notation $A \geq B$ means that the difference matrix $A - B$ is positive semi-definite). By a standard property of partitioned positive semi-definite matrices ([52]), equation (7.17) is equivalent to:

$$\begin{bmatrix} P^{-1} & B^* \\ B & R \end{bmatrix} = \begin{bmatrix} P^{-1/2} \\ BP^{1/2} \end{bmatrix} \begin{bmatrix} P^{-1/2} & P^{1/2} B^* \end{bmatrix} \geq 0 \quad (7.18)$$

which is obviously true. The result (7.15) is therefore proved. Furthermore, it can be easily verified that the minimizing vector β is given by:

$$\beta = PB^*R^{-1}y \quad (7.19)$$

a fact that will be used in the next sub-section (note that (7.19) is the MAP-like estimate of β in (7.7), which was mentioned in the previous discussion).

Making use of the observation that the solution p to (7.10) is identical to the minimizer p of (7.13), we will obtain this solution from (7.13); interestingly the latter problem can be solved more efficiently than (7.10), as explained in the following, in spite of the additional variable β in (7.13).

The minimization of (7.13) wrt p , for fixed β , decouples in $(M+N)$ one-dimensional problems with the following generic form :

$$\min_{p \geq 0} \frac{|\beta|^2}{p} + w^2 p \quad (7.20)$$

or, equivalently,

$$\min_{p \geq 0} \left(\frac{|\beta|}{\sqrt{p}} - w\sqrt{p} \right)^2 + 2w|\beta|. \quad (7.21)$$

It follows easily from (7.21) that the minimizer p of (7.13), for fixed β , is:

$$p_k = \frac{|\beta_k|}{w_k} \quad k = 1, \dots, M+N \quad (7.22)$$

and also that the minimization problem wrt β that remains to be solved is given by

$$\begin{aligned} \min_{\beta} & \sum_{k=1}^{M+N} w_k |\beta_k| \\ \text{s.t. } & B\beta = y. \end{aligned} \quad (7.23)$$

This problem can be readily cast as an SOCP [16] (with $\{\alpha_k\}$ being auxiliary variables):

$$\begin{aligned} & \min_{\{\alpha_k\}, \beta} \sum_{k=1}^{M+N} w_k \alpha_k \\ & \text{s.t. } |\beta_k| \leq \alpha_k \quad k = 1, \dots, M+N \\ & B\beta = y \end{aligned} \tag{7.24}$$

which can be solved much more efficiently than the SDP in (7.12).

In summary, the SOCP-based SPICE algorithm consists of solving the SOCP in (7.24) to obtain an estimate of β (and hence of x). The solution p to the original SPICE problem in (7.10), if desired, can be obtained from β via (7.22).

It is worth noting that in the case of *real-valued data*, the SOCP in (7.24) reduces to the following *linear program* (LP) (below, $\alpha_k, \beta_k \in \mathbb{R}$):

$$\begin{aligned} & \min_{\{\alpha_k, \beta_k\}} \sum_{k=1}^{M+N} w_k \alpha_k \\ & \text{s.t. } -\alpha_k \leq \beta_k \leq \alpha_k; \alpha_k \geq 0; \quad k = 1, \dots, M+N \\ & B\beta = y \end{aligned} \tag{7.25}$$

which can be solved quite efficiently. In effect there are a host of super-fast algorithms in the literature that can be used to solve (7.25), such as homotopy-based methods and iterative thresholding-based algorithms (see, e.g., [29]).

For real-valued data, the use of such an efficient algorithm to solve the LP in (7.25) is currently the fastest available method to compute the SPICE estimate. However, in the complex-valued data case, the cyclic algorithm (CA) presented in the next sub-section can be a faster way of solving (7.13) than the SOCP-based algorithm of this section.

7.2.2 CA-based solver

The minimizers of the criterion in (7.13) wrt β , for fixed p , and wrt p , for fixed β , have been derived in the previous sub-section : they are given by the closed-form expressions in (7.19) and, respectively, (7.22). Therefore, the main ingredients of a CA for the minimization of (7.13) are already available. Let the super-index i indicate the i -th iteration, and let $P(i)$ denote the matrix P made from $\{p_k^i\}$. The CA-based SPICE algorithm consists of the following equations (which are to be iterated until a convergence criterion is satisfied) :

$$\begin{aligned} p_k^i &= |\beta_k^i| / w_k \\ \beta_k^{i+1} &= p_k^i b_k^* R^{-1}(i) y; R(i) = B P(i) B^* \end{aligned} \tag{7.26}$$

($k = 1, \dots, M+N$; $i = 0, 1, 2, \dots$). This iterative algorithm *converges globally* to the solution of (7.13) from any initial values $\{p_k > 0\}$ or $\{\beta_k \neq 0\}$ (see [93] and the references there). However, needless to say, the rate of convergence may depend on the initial value. Here we will initialize (7.26) with the

element-wise least-squares estimate of β in (7.1) :

$$\beta_k^0 = \frac{b_k^* y}{\|b_k\|^2} \quad k = 1, \dots, M+N. \quad (7.27)$$

Note that in the case of the SOCP-based SPICE the user does not need to choose any initial values, as the SOCP solver selects them implicitly.

We have compared the execution times of the CA-based and the SOCP-based algorithms for computing the SPICE estimate in a number of complex-valued data cases. It is our experience that the CA-based algorithm can be faster than the SOCP-based one for small to medium values of M , whereas it tends to be slower for large values of M . Because we have used a state-of-the art solver for SOCP whereas our Matlab code for CA is likely far from optimal (in particular this code comprises several loops which are notoriously slow in Matlab), we decided to include neither run time plots for the two algorithms nor any specific indication as to the value of M beyond which the SOCP-based solver becomes faster than the CA-based algorithm : after all, such a value of M should depend not only on the codes used for the two algorithms and on the machine on which they are run but also on the problem under consideration (in particular on the matrix A of that problem).

7.3 LIKES

We will now make the additional assumption that x and e are circularly Gaussian distributed. This means that y has a circular Gaussian distribution with zero mean and covariance matrix equal to R . Consequently the negative log-likelihood function associated with y is given (to within an additive constant) by:

$$f(p) = \ln|R| + y^* R^{-1} y. \quad (7.28)$$

We will obtain an estimate of p by minimizing this function, i.e. :

$$\min_p y^* R^{-1} y + \ln|R|. \quad (7.29)$$

The objective in (7.29) is a well-established fitting criterion even when the data are *not* Gaussian distributed. In fact, in the one-snapshot case considered in this chapter, (7.29) may be deemed to be statistically a more appealing estimation criterion than the covariance fitting metric in (7.10).

With regard to the form of the two fitting criteria in (7.10) and (7.29), we see that they share a common term, viz. $y^* R^{-1} y$, which is a convex function of p . The problem is that the second term in (7.29), unlike that in (7.10), is not convex. In fact this term can be shown to be a *concave function of p* see, e.g., [119] [16]; while this property is known, we include a proof of it in 7.5 to make the chapter as self contained as possible. The consequence of this fact is, as shown in 7.6, that (7.29) is a *non-convex problem* which, unlike the

SPICE problem in the previous section, may be hard to solve globally. In the following we will derive an iterative algorithm for the minimization problem in (7.29), which *decreases the criterion $f(p)$ at each iteration* and can thus be expected to converge at least locally. As we will see shortly, each iteration of this new algorithm turns out to require solving a SPICE-like problem, which can be done by means of the SPICE solvers presented in the previous section.

Let \tilde{p} be an arbitrary point in the parameter space, and let \tilde{R} denote the corresponding covariance matrix. Because a concave function is majorized by its tangent plane at any point, the following inequality must hold for any p :

$$\begin{aligned} \ln|R| &\leq \ln|\tilde{R}| + \sum_{k=1}^{M+N} \text{tr} \left(\tilde{R}^{-1} b_k b_k^* \right) (p_k - \tilde{p}_k) \\ &= \ln|\tilde{R}| - N + \text{tr}(\tilde{R}^{-1} R) = \ln|\tilde{R}| - N + \sum_{k=1}^{M+N} \tilde{w}_k^2 p_k \end{aligned} \quad (7.30)$$

where

$$\tilde{w}_k^2 = b_k^* \tilde{R}^{-1} b_k. \quad (7.31)$$

It follows from (7.30) that :

$$f(p) \leq (\ln|\tilde{R}| - N) + y^* R^{-1} y + \sum_{k=1}^{M+N} \tilde{w}_k^2 p_k \triangleq g(p) \quad (7.32)$$

for any vectors \tilde{p} and p . Note also that

$$f(\tilde{p}) = g(\tilde{p}). \quad (7.33)$$

The important implication of (7.32) and (7.33) is that we can decrease the function $f(p)$ from $f(\tilde{p})$ to, let us say, $f(\hat{p})$ by choosing \hat{p} as the minimum point of $g(p)$ or at least such that $g(\hat{p}) > g(\tilde{p})$:

$$f(\hat{p}) \leq g(\hat{p}) < g(\tilde{p}) = f(\tilde{p}) \quad (7.34)$$

(the first inequality above follows from (7.32), the second inequality from the definition of \hat{p} , and the equality from (7.33).) This is in fact the underlying principle of the minimization-majorization approach to solving a given minimization problem, see, e.g. [101] [46]. The usefulness of the said approach depends on whether the minimization (or the decrease) of $g(p)$ is easier than that of $f(p)$. Here this is definitely true, as $g(p)$ is (to within a constant) a SPICE-like convex criterion function, compare it with (7.10). Consequently, the SPICE solvers of the previous section can be used to find a vector \hat{p} with the above property, for any given \tilde{p} .

The procedure for the minimization of the negative log-likelihood function in (7.29), outlined above, is designated by the acronym LIKES (**L**Ikehood-based **E**stimation of **S**parse parameters). LIKES consists of an initialization stage (we can use the same initial values as for SPICE, see (7.27), to compute the initial weights $\{\tilde{w}_k\}$; alternatively we can set $\{\tilde{w}_k = w_k\}$ where $\{w_k\}$ are

the SPICE weights in (7.11)), and of the following main steps that are to be iterated until a convergence criterion is satisfied :

Inner step. Using the most recent estimate to define \tilde{p} , build \tilde{R} and employ either of the SPICE solvers to obtain the next estimate \hat{p} (as well as $\hat{\beta}$, if desired). Note that an off-the-shelf SOCP solver will compute \hat{p} as the minimum point of $g(p)$; on the other hand, if desired, the CA-based solver can be iterated only as long as it decreases $g(p)$ “significantly” and therefore it can be stopped before complete convergence to speed up the computation of \hat{p} ; the trade-off being that a presumably larger number of outer iterations may then be required.

Outer step. Set $\tilde{p} = \hat{p}$, and go to the inner step.

It is somewhat interesting to note that neither of the estimation methods presented in this chapter appears to make *explicit* use of the information that the parameter vector is sparse, and yet both SPICE and LIKES provide sparse estimates (see the next section). To understand this behavior, at least partly, observe that the second term in the p -dependent SPICE criterion in (7.10) is nothing but the (weighted) ℓ_1 -norm of the parameter vector, which is known to penalize non-sparse vectors ; and the same is true for the β -dependent SPICE criterion in (7.23) where the objective function itself is the weighted ℓ_1 -norm of β . These observations also apply to LIKES whose inner step is a SPICE-like estimation problem.

We expect that the LIKES estimates of β and p are more accurate than the SPICE estimates. A reason for this expectation is the maximum-likelihood character of LIKES. The fact that the weights $\{\tilde{w}_k\}$ of LIKES are *adaptive* also makes LIKES more appealing than SPICE whose weights $\{w_k\}$ are constant : indeed, \tilde{w}_k^2 can be interpreted as an approximation of the inverse power $1/\tilde{p}_k$ (see [98]) and thus if \tilde{p}_k was “small” then \hat{p}_k is likely to be even smaller; in particular, this means that the LIKES estimate can be expected to be sparser than the SPICE estimate. The numerical results presented in the next section lend support to the expected better accuracy of LIKES. On the other hand, we should note that LIKES is computationally more demanding than SPICE. Roughly speaking, the execution time of LIKES is equal to that of SPICE times the number of run outer-step iterations (which appears to be usually between 10 and 20).

Next we comment on the relationship between LIKES and the sparse bayesian learning (SBL) approach of [109] and [119]. While both LIKES and SBL make use of the maximum-likelihood principle, as in (7.29), the two algorithms are quite different from one another : SBL is an expectation-maximization (EM) - type of method, whereas LIKES is a minimization-majorization (MM) - based technique (as explained above). In general EM algorithms are known to converge more slowly than MM counterparts (see, e.g., [46]) and, indeed, in a number of preliminary tests we have observed that LIKES can converge at a much faster rate than SBL. Consequently, from a pragmatic/computational perspective, LIKES may be preferable to the SBL algorithm of [119] [109] (a

detailed comparison of LIKES and SBL is left to future work). From a theoretical viewpoint, on the other hand, the analysis in [119] and [109] has a number of interesting results that also apply to LIKES *mutatis mutandis*. In particular, the cited papers (especially [119]) contain a detailed explanation of the sparsity of the LIKES/SBL parameter estimates : in a nutshell, if $\{p_k\}_{k=1}^M$ in (7.3) are assumed to have a certain prior then the resultant hierarchical prior for x can be shown to have a sparsity-inducing character. Furthermore, [119] includes an useful analysis of the global and local minima of the negative log-likelihood function in (7.28) which, in the case of local minima, holds under either noisy or noiseless conditions.

Finally we end this section with a discussion on the similarities and differences between LIKES and the related algorithms of [90] [117]. The operational principles of LIKES and the algorithm in [90] are quite similar but the data model considered in the cited paper is different from (7.1) and thus the implementation details of the two algorithms are rather different from one another. Regarding the algorithm in [117], once again the basic operating principles are similar to each other but there exist important differences as well. In particular, [117] assumes the special case of $\sigma_1 = \dots = \sigma_N \stackrel{\Delta}{=} \sigma$ and also that σ is given. This assumption leads to an optimization problem that is somewhat different from (7.23); furthermore, unlike what we did in Section 7.2.2, [117] does not derive any specific algorithm for solving such a problem.

7.4 Numerical illustrations and concluding remarks

7.4.1 Spectral analysis example

We consider the problem of estimating the amplitudes $\{c_k\}$ and frequencies $\{\omega_k\}$ of three sinusoidal signals from noisy irregularly-sampled observations :

$$y(t_n) = \sum_{k=1}^3 c_k e^{i\omega_k t_n} + e(t_n) \quad ; n = 1, \dots, 50 \quad (7.35)$$

where the noise samples $\{e(t_n)\}$ are uncorrelated Gaussian random variables with zero mean and variance σ , and the sampling times $\{t_n\}$ are drawn from a uniform distribution over the interval $[0, 100]$. The true values of the sinusoidal parameters in (7.35) are the following ones :

$$\begin{aligned} c_1 &= 5 & c_2 &= 5 & c_3 &= 10 \\ \omega_1 &= 2\pi 0.3 & \omega_2 &= 2\pi 0.5 & \omega_3 &= 2\pi 0.52. \end{aligned} \quad (7.36)$$

The signal-to-noise ratio (SNR) for (7.35)-(7.36) is defined as :

$$\text{SNR} = 10 \log(100/\sigma). \quad (7.37)$$

Let

$$a_p = \begin{bmatrix} e^{iv_p t_1} \\ \vdots \\ e^{iv_p t_N} \end{bmatrix} ; p = 1, \dots, M \quad (7.38)$$

where

$$v_p = \frac{2\pi}{M} p ; N = 50 ; M = 1000. \quad (7.39)$$

Using the above notation we can re-write (7.35) in the form (7.1), viz.

$$y = Ax + e \quad (7.40)$$

with the parameter vector x having only three non-zero elements which are equal to $\{c_k\}_{k=1}^3$. Herein we would like to note that the true frequencies $\{\omega_k\}_{k=1}^3$ were assumed here to lie on the grid $\{v_p\}_{p=1}^{1000}$ however, the off-grid case can be treated similarly with some slight modifications in the data model see, e.g. [91] for more details on it.

The SPICE and LIKES estimates of $\{|x_k|\}$, obtained in 100 Monte-Carlo runs using the SOCP-based solver with non-identical $\{\sigma_k\}$, are shown (in a superimposed manner) in Figure 7.1 along with the IRL1 and BP estimates (for SNR = 15dB). The latter two estimates require knowledge of the noise variance σ . In applications σ must, of course, be estimated from y and this is not an easy task. In order to estimate σ as reliably as possible we assume that *we know the sparsity of x* in (7.40) (i.e. the number of non-zero elements of x). While this appears to be a less restrictive assumption than assuming that σ is known (as done in the cited papers where IRL1 and BP were introduced), it is still a relatively impractical assumption and making it basically eliminates one of the main advantages of the sparse estimation methods over the parametric ones (indeed, once we know that the number of components in (7.35) is three we can apply a powerful parametric method to y to estimate the signal parameters, see, e.g. [98]).

To estimate σ we make use of the initial estimate of x in (7.27), viz. $|x_k| = |a_k^* y| / \|a_k\|^2$ (for $k = 1, \dots, M$), which is nothing but the Periodogram. We estimate the frequencies $\{\omega_k\}_{k=1}^3$ as the locations of the three largest peaks of $\{|x_k|\}$ and then use these frequency estimates in (7.35) to estimate $\{c_k\}_{k=1}^3$ via least squares. Finally we obtain an estimate of σ as the sample variance of the residuals of the said least-squares problem.

All four estimates shown in Figure 7.1 are sparse but this is not fully obvious for SPICE and LIKES due to overlaying many realizations and to the fact that the zero elements in the estimated parameter vector may appear at different positions in various realizations; to shed some light on this aspect, in Figure 7.2 we show one randomly selected plot from each of Figures 7.1 a) - d). We note the following facts from Figures 1 and 2 : the LIKES estimate is more accurate than SPICE and both outperform IRL1 and BP. The latter two estimates are slightly biased for frequencies (see the insets in Figure 1) and

heavily biased downward for amplitudes presumably due to an overestimation of σ ; we remind the reader that σ is estimated under the practically restrictive assumption that the number of sinusoids in the data is known : without this assumption simple estimation of σ is likely to be even less accurate and the performance of IRL1 and BP to degrade accordingly.

The three largest elements of the estimated x are taken as estimates of $\{c_k\}_{k=1}^3$. The MSE's of these estimates, obtained from 100 Monte-Carlo simulation runs, are shown in Figure 7.3 for $\text{SNR} \in [0\text{dB}, 25\text{dB}]$. We observe from this figure that LIKES and SPICE significantly outperform IRL1 and BP. Note also from this figure that the LIKES estimate is the most accurate one for all SNR values considered.

Finally we note that, once the frequencies $\{\omega_k\}$ are determined, e.g. by either SPICE or LIKES (see [93] for histogram plots illustrating the accuracy of SPICE frequency estimates), the amplitudes $\{c_k\}$ could be estimated by least squares (LS). However, as already mentioned above when discussing the estimation of σ for IRL1 and BP, LS is a parametric method that estimates only the amplitudes of the three sinusoids with given frequencies $\{\omega_k\}$ and therefore its use has to be preceded by a step for estimating the number of sinusoids in the data. In contrast with this, SPICE and LIKES are *non-parametric* methods which do *not* require such a step (note, for instance, that even when we showed the MSE's of the estimated amplitudes in Figure 7.3 we estimated *all* $\{x_k\}$ not only the three of them having the largest magnitude). For this reason, we do not consider the LS amplitude estimate here but note that this estimate is quite competitive if we decide to go parametric and dispose of a good detector for estimating the number of signals in the observed data.

7.4.2 Range-Doppler imaging example

In this sub-section we consider a sensing system (such as radar) that transmits a probing signal towards an area of interest. Let $\{s_k\}_{k=1}^N$ denote the code used to modulate the transmitted waveform, which is assumed to be a pseudo-random sequence with zero mean and unit variance. Then a simplified (non-parametric) model for the received signal (after demodulation and a number of other pre-processing operations) is as follows (e.g. [84]):

$$y = \sum_{k=-N+1}^{N-1} \sum_{l=-L}^{L-1} x_{kl} a_{kl} + e. \quad (7.41)$$

In (7.41), x_{kl} is the reflectivity coefficient for the cell corresponding to the k -th range bin and the l -th Doppler bin, e is a white noise with zero mean and variance σ , and

$$a_{kl} = J_k \begin{bmatrix} s_1 e^{i\omega_l} \\ \vdots \\ s_N e^{iN\omega_l} \end{bmatrix} \quad (7.42)$$

where ω_l is the frequency associated with the l -th Doppler bin,

$$\omega_l = \frac{\pi}{L} l, \quad (7.43)$$

and J_k denotes the following shifting matrix

$$J_k = \begin{bmatrix} 0 & \cdots & \overbrace{1}^{k+1} & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} = J_{-k}^* \quad k = 0, \dots, N-1. \quad (7.44)$$

We consider the case of five targets present in (7.41), with the following parameters:

Range bin k :	0	0	0	2	-2
Doppler bin l :	-15	5	10	-15	20
Reflectivity coefficient x_{kl} :	5	5	10	5	5

(7.45)

Similarly to (7.37), we define the SNR as

$$\text{SNR} = 10 \log(100/\sigma). \quad (7.46)$$

We will show results for a data realization with $\text{SNR} = 20\text{dB}$, $N = 50$, and $L = 25$.

Note that the data vector y in (7.41), which is temporally aligned with the 0-th range bin, is typically used to estimate the parameters of the targets (i.e. the reflectivities and Doppler shifts) that are present only in the 0-th range bin. Indeed, targets in the k -th range bin are “observable” in fewer and fewer elements of y as $|k|$ increases and therefore their estimation from y in (7.41) might not be reliable; to estimate the parameters of such targets with a satisfactory accuracy we need another segment of the entire received data string, which is properly aligned with the range bin of interest.

With the above fact in mind we have placed the existing targets in the central range bin $k = 0$ and in two bins close to it $k = \pm 2$, see (7.45). Nevertheless, we will not estimate $\{x_{kl}\}$ only for these three values of k , but will use the data vector y to estimate *all* $\{x_{kl}\}$.

Evidently the data model (7.41) has the same form as (7.1) (with $M + N = 5000$) and hence we can directly apply the initial estimator in (7.27) as well as SPICE and LIKES to it; note, once again, that the initial estimate is the Periodogram (which is still the method preferred by practitioners for this type of applications). The so-obtained estimates of $\{|x_{kl}|\}$, also known as the estimated range-Doppler images, along with the true image are shown in Figure

7.4 where $|x_{kl}|$ occurs at position (k, l) . Both SPICE and LIKES yield precise estimates of the true range-Doppler image; in particular, observe the considerable sparsity of SPICE and LIKES images which are not affected by noise even for range bins far away from the central one (the images obtained by IRL1 and BP have a poorer performance than the SPICE and LIKES images and are thus omitted). In contrast to this, the Periodogram estimate is rather noisy and as a consequence most targets are barely visible if at all; in fact this estimate was so distorted for $k \notin [-20, 20]$ that we chose to present the Periodogram image in Figure 7.4 only for $k \in [-20, 20]$.

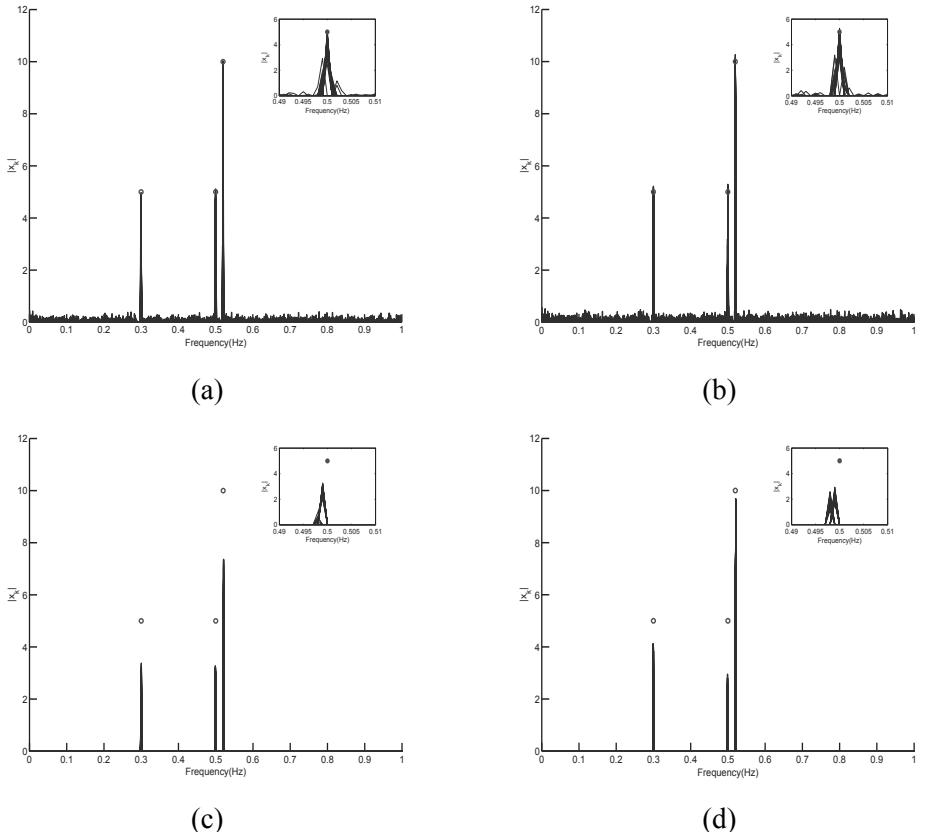
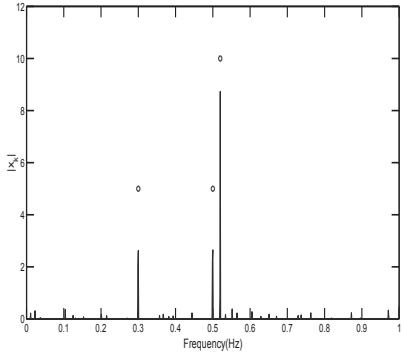
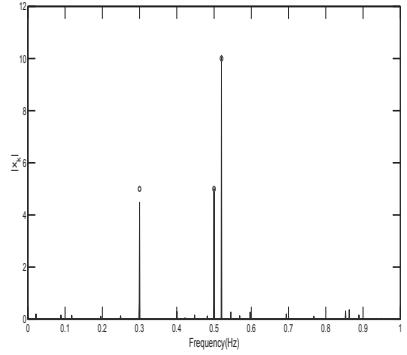


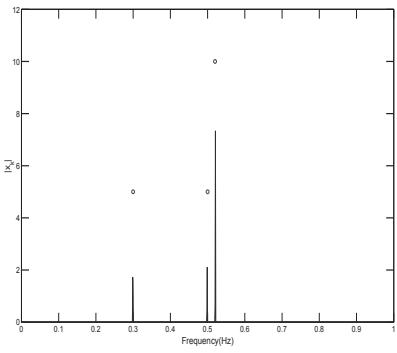
Figure 7.1. Superimposed plots of $\{ |x_k| \}$ obtained via (a) SPICE (b) LIKES (c) BP and (d) IRL1 in 100 Monte-Carlo runs. The circles indicate the true parameter values. The zoom-in plots show the spectrum in the interval $[0.49 - 0.51]$ Hz



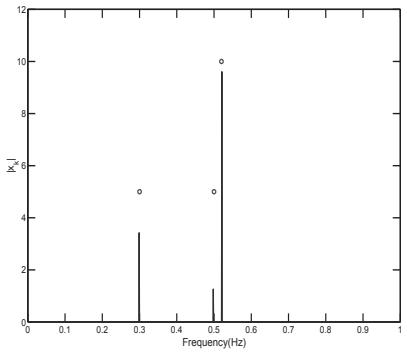
(a)



(b)

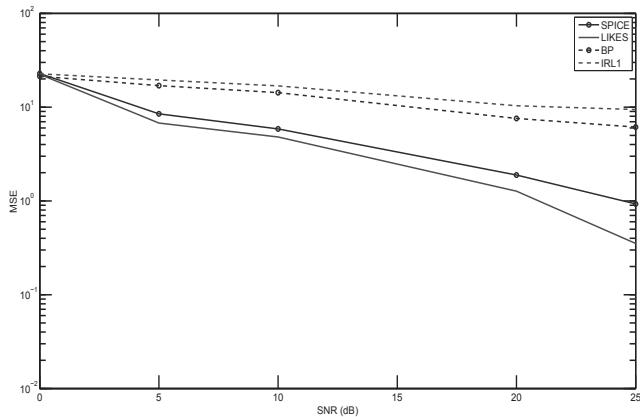


(c)

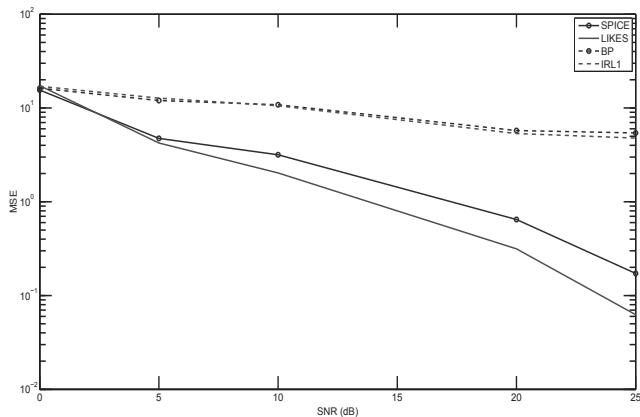


(d)

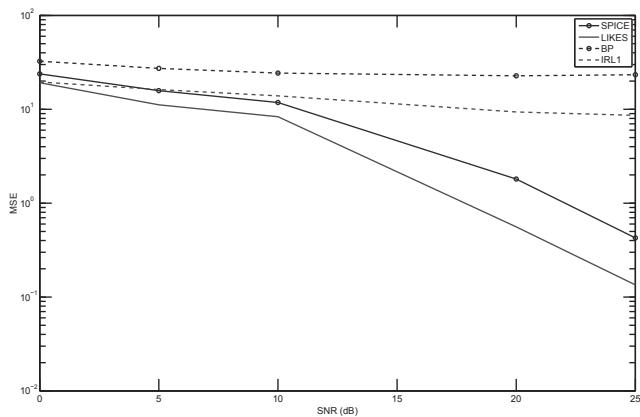
Figure 7.2. Four randomly selected plots of $\{|x_k|\}$ from Figure 7.1 for (a) SPICE (b) LIKES (c) BP and (d) IRL1.



(a) c_1

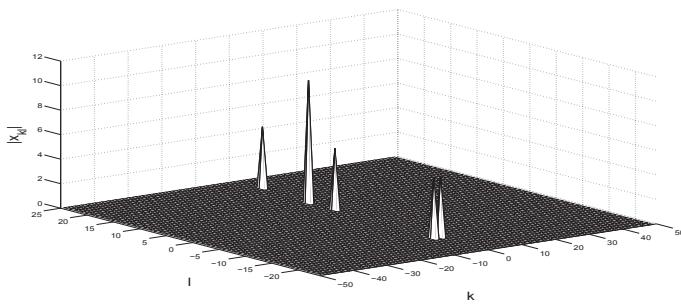


(b) c_2

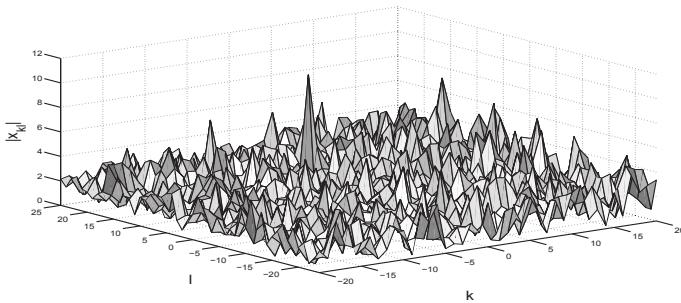


(c) c_3

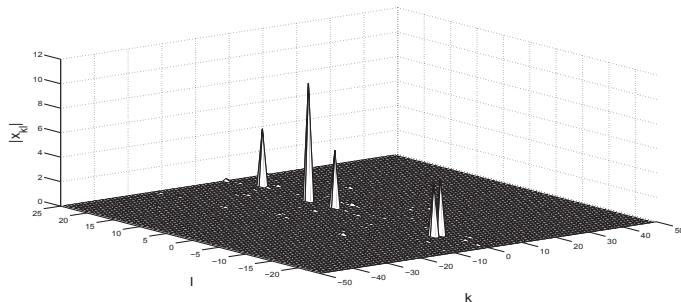
Figure 7.3. MSE vs SNR for the SPICE, LIKES, BP and IRL1 estimates of $\{c_k\}_{k=1}^3$.
148



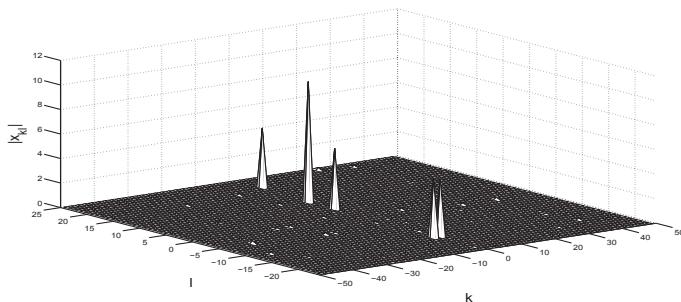
(a) True



(b) Periodogram



(c) SPICE



(d) LIKES

Figure 7.4. The true and estimated range-Doppler images.

7.5 Concavity proof

We prove here that $\ln|R|$ is a concave function of p by showing that its Hessian matrix is negative semi-definite at any point in the parameter space. We have that :

$$\frac{\partial \ln|R|}{\partial p_k} = \text{tr} \left(R^{-1} \frac{\partial R}{\partial p_k} \right) = b_k^* R^{-1} b_k \quad (7.47)$$

and

$$\frac{\partial^2 \ln|R|}{\partial p_k \partial p_s} = -b_k^* R^{-1} b_s b_s^* R^{-1} b_k \stackrel{\Delta}{=} -H_{ks}. \quad (7.48)$$

The matrix H , introduced above, must therefore be shown to be positive semi-definite. In other words we have to prove that $g^* H g \geq 0$ for any vector $g = [g_1, \dots, g_{M+N}]^T$. Let

$$X = \sum_{k=1}^{M+N} g_k b_k b_k^* \quad (7.49)$$

and let $\text{vec}(X)$ denote the vector made from the columns of X stacked on top of each other. Then a simple calculation shows that:

$$\begin{aligned} g^* H g &= \sum_{p=1}^{M+N} \sum_{s=1}^{M+N} g_p^* g_s \text{tr} (R^{-1} b_p b_p^* R^{-1} b_s b_s^*) = \text{tr} (R^{-1} X^* R^{-1} X) \\ &= \text{vec}^*(X) (R^{-T} \otimes R^{-1}) \text{vec}(X) \end{aligned} \quad (7.50)$$

where \otimes denotes the Kronecker matrix product. Because the matrix $R^{-T} \otimes R^{-1}$ is positive definite, it follows from (7.50) that $g^* H g \geq 0$ and the proof is concluded.

7.6 Non-convexity proof

The Hessian matrix associated with the function $f(p)$ in (7.28) has the following elements :

$$\begin{aligned} \frac{\partial^2 f(p)}{\partial p_k \partial p_s} &= -b_k^* R^{-1} b_s b_s^* R^{-1} b_k + 2\text{Re}(y^* R^{-1} b_k b_k^* R^{-1} b_s b_s^* R^{-1} y) \\ &\stackrel{\Delta}{=} -H_{ks} + G_{ks} \quad k, s = 1, \dots, M+N \end{aligned} \quad (7.51)$$

where $\text{Re}(z)$ denotes the real part of z . The first term in (7.51) follows from (7.48) and it corresponds to a negative semi-definite matrix (see the previous appendix); the second term in the above equation, which corresponds to the Hessian matrix of $y^* R^{-1} y$, can be similarly shown to be positive semi-definite : $G \geq 0$. For sufficiently small values of $\|p\|$, G dominates H in (7.51); whereas the opposite is true if $\|p\|$ is large enough. It follows that the Hessian matrix in (7.51) becomes positive semi-definite as $\|p\|$ approaches zero, comes to be negative semi-definite as $\|p\|$ approaches infinity, and in general is indefinite for some intermediate values of $\|p\|$. With this observation, the proof is concluded.

7.7 Connection between SPICE and the square-root Lasso

Consider the augmented problem

$$\begin{aligned} \min_{p, \beta} & \beta^* P^{-1} \beta + \sum_{k=1}^{M+N} w_k^2 p_k \\ \text{s.t. } & B\beta = y. \end{aligned} \quad (7.52)$$

If we assume that $p_{M+1} = p_{M+2} = \dots = p_{M+N} = \sigma$ - this is equivalent to assuming that the noises in the different data samples have the same variance, then the minimizers p and σ of (7.52) for fixed β , are given by:

$$\begin{aligned} p_k &= \frac{|\beta_k|}{w_k} \quad k = 1, \dots, M \\ \sigma &= \frac{\sqrt{|\beta_{M+1}|^2 + |\beta_{M+2}|^2 + \dots + |\beta_{M+N}|^2}}{\sqrt{w_{M+1}^2 + w_{M+2}^2 + \dots + w_{M+N}^2}} \\ &= \frac{\sqrt{|\beta_{M+1}|^2 + |\beta_{M+2}|^2 + \dots + |\beta_{M+N}|^2}}{\sqrt{N}} \end{aligned} \quad (7.53)$$

and the minimization problem wrt β becomes:

$$\begin{aligned} \min_{\beta} & \sum_{k=1}^M w_k |\beta_k| + \sqrt{N} \sqrt{|\beta_{M+1}|^2 + |\beta_{M+2}|^2 + \dots + |\beta_{M+N}|^2} \\ \text{s.t. } & B\beta = y. \end{aligned} \quad (7.54)$$

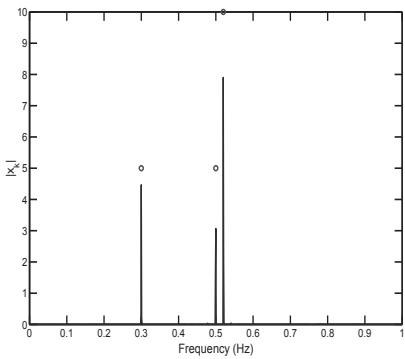
If we assume that $\{w_k^2\}_{k=1}^M = N$, which is in fact the case in the spectral analysis example considered in the numerical section, then (7.54) becomes:

$$\begin{aligned} \min_{\beta} & \sum_{k=1}^M |\beta_k| + \sqrt{|\beta_{M+1}|^2 + |\beta_{M+2}|^2 + \dots + |\beta_{M+N}|^2} \\ \text{s.t. } & B\beta = y. \end{aligned} \quad (7.55)$$

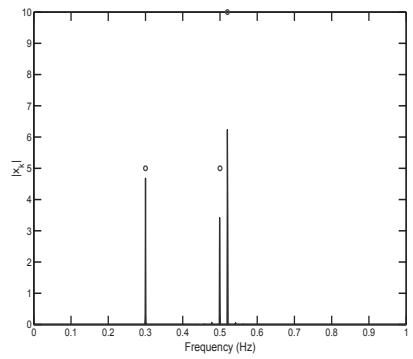
which relates to the square-root Lasso algorithm proposed recently in [10] except for the difference that in the square-root Lasso algorithm there is a constant factor c in the first term of the cost function; more precisely, the square-root Lasso algorithm is given by:

$$\begin{aligned} \min_{\beta} & c \sum_{k=1}^M |\beta_k| + \sqrt{|\beta_{M+1}|^2 + |\beta_{M+2}|^2 + \dots + |\beta_{M+N}|^2} \\ \text{s.t. } & B\beta = y. \end{aligned} \quad (7.56)$$

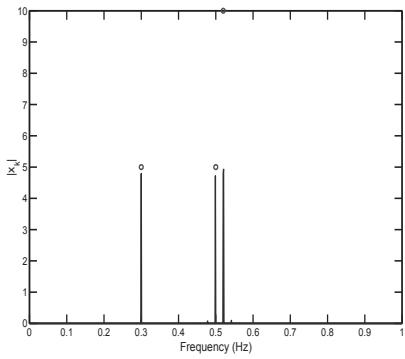
Moreover, in [10] it is shown that the choice of c is independent of the noise variance and c should lie within the interval $[1, 4]$. Thus, SPICE corresponds to the square-root Lasso algorithm with $c = 1$. In Figure 7.5, we show the square-root Lasso estimates of $\{|x_k|\}$ for four different values of c chosen within the interval $[1, 4]$ for the spectral analysis example considered in the numerical section. It can be seen from the figure that the estimates corresponding to different values of c find the positions of the true frequencies exactly but with different estimates of amplitudes.



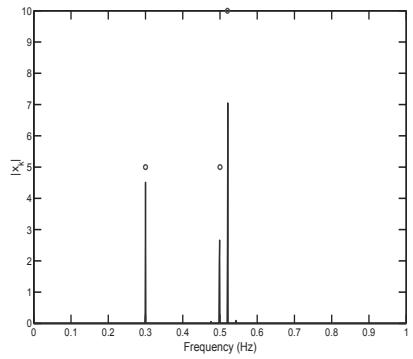
(a)



(b)



(c)



(d)

Figure 7.5. $\{|x_k|\}$ of square-root Lasso algorithm for the choice of (a) $c = 1$ (which corresponds to SPICE) (b) $c = 2$ (c) $c = 3$ and d) $c = 4$, SNR = 15 dB

8. On Grid selection problems and their solutions for sparse estimation of spectral lines

8.1 Introduction and problem formulation

We will consider the case of temporal spectral analysis (see, e.g., [93] [80] [121]) but the ideas and methods presented in what follows can be used in a direct manner for spatial spectral analysis as well (e.g. [92] [98]). Let

$$y = [y(t_1), \dots, y(t_N)]^T \in \mathbb{C}^{N \times 1} \quad (8.1)$$

denote the data vector, where $\{t_k \in \mathbb{R}^+\}$ are the observation times, and assume that y satisfies the following spectral line model (also called a sinusoids-in-noise model):

$$y = \sum_{p=1}^n \alpha_p a(f_p) + \varepsilon \quad (8.2)$$

where $\varepsilon \in \mathbb{C}^{N \times 1}$ is the noise vector (assumed to have independent elements), $\{\alpha_p \in \mathbb{C}\}$ are the amplitudes of the n sinusoidal components, $\{f_p \in \mathbb{R}\}$ are their frequencies, and

$$a(f) = [e^{i2\pi f t_1}, \dots, e^{i2\pi f t_N}]^T \in \mathbb{C}^{N \times 1} \quad (8.3)$$

The problem is to estimate the parameters in (8.2), viz. $\{\alpha_p, f_p\}$, from y . This can be done by a multitude of methods, see, e.g. the review paper [6]. A class of such methods is that of the parametric approaches, such as nonlinear least squares methods, a drawback of which is that they require a guess of n before they can be applied to y . The parameters in (8.2) can also be estimated by means of nonparametric approaches, such as the periodograms, whose use does not require knowledge of n . However most of the latter approaches suffer from low performance problems, such as a rather limited resolving power. A third class of possible approaches, which may be called semi-parametric (see, e.g., [93]), is that of sparse estimation methods. The performance of these methods is somewhat similar to that of the parametric approaches (in particular their resolving power can surpass that of the basic periodogram) and yet their use does not necessarily require apriori knowledge of n (similarly to the nonparametric approaches). These two attractive features have led in recent years to a significant interest in sparse estimation methods for spectral line

analysis, somewhat similar to the excitement about subspace-based parametric methods in the 80's. However, on their turn, the sparse estimation methods are *not* free of problems and therefore they are not necessarily preferable to either parametric or non-parametric approaches. To explain these problems in sufficient detail, let f_{\max} be an upper bound on $\{f_p\}$, that is $f_{\max} \geq f_p$ (for $p = 1, \dots, n$) and let Δ denote the step size of a uniform grid covering the interval $[0, f_{\max}]$ (for notational simplicity we will consider only positive frequencies but negative frequencies can be dealt with in the same way); note that the choice of f_{\max} and Δ will be addressed in Section 8.2. Also let K be the number of points of the said grid :

$$K = \lfloor f_{\max}/\Delta \rfloor \quad (8.4)$$

Finally, let

$$A = [a(0), \dots, a((K-1)\Delta)] \in \mathbb{C}^{N \times K}. \quad (8.5)$$

We consider the grid bins to be defined as

$$I_k = [\mu_k, \mu_{k+1}]; \mu_k = -\Delta/2 + (k-1)\Delta, k = 1, \dots, K \quad (8.6)$$

so that the frequencies appearing in A , viz.

$$v_k = (k-1)\Delta, k = 1, \dots, K \quad (8.7)$$

are the middle points of the bins.

Using the above notation we can write (8.2), at least approximately, in the following form :

$$y = A\beta + \epsilon \quad (8.8)$$

where $\beta = [\beta_1, \dots, \beta_K]^T$ is ideally a sparse vector with many zero elements and whose non-zero elements are equal to $\{\alpha_p\}$ and correspond to frequencies in A that are equal to $\{f_p\}$. The estimation of the amplitudes and frequencies in (8.2) can therefore be reduced to the *sparse estimation* of β in (8.8) and to the detection of the non-zero elements of this vector.

There are a host of methods for sparse estimation but none of them is free of problems as we now explain. *First* of all, most of these methods require prior information on the noise variance, or the number of non-zero components of β and the like (see, e.g., [61]). Such information is usually unavailable in applications and obtaining it is not an easy task. LIKES (**L**I**K**eihood-based **E**stimation of **S**parse parameters), which is the sparse estimation method used in this work, is one of the few existing methods that do not require the above type of prior information and is thus straightforward to use (see [92]); note that LIKES is related to the Sparse Bayesian Learning (SBL) method of [109] and [119], see [92] for details. *Second*, and definitely not less important, all sparse estimation methods (LIKES included) require grid selection which essentially reduces to the choice of K . In spite of a growing literature on this

problem (see, e.g., [2]), there is apparently no satisfactory solution to it: most proposed solutions are either ad-hoc or too computationally intensive or both. Our main goal here is to correct this situation by introducing a conceptually simple method for solving the grid selection problem and its derivatives. Note that, while we focus the following discussion on the use of LIKES, the proposed grid selection approach applies *mutatis mutandis* to virtually any other sparse estimation method.

8.2 Grid selection : preliminary ideas

8.2.1 Practical guideline

Let

$$\tilde{f}_{max} = \frac{1}{2 \min_{k \in [1, N-1]} (t_{k+1} - t_k)} \quad (8.9)$$

There is empirical evidence that usually one may conduct alias-free spectral analysis for frequencies up to \tilde{f}_{max} but not beyond (see, e.g., [31] [93]). Consequently f_{max} introduced before must be such that

$$f_{max} \leq \tilde{f}_{max} \quad (8.10)$$

(we can choose f_{max} to be strictly less than \tilde{f}_{max} if we have some information about the range of $\{f_p\}$ in (8.2)). Furthermore, $1/(t_N - t_1)$ is known to be the resolution limit of the periodogram (in the sense that, with a high probability, the periodogram cannot resolve two sinusoids separated in frequency by less than $1/(t_N - t_1)$ [93] [98]). The sparse estimation methods are expected to have a better resolving power than this. As a consequence, the bin size of the frequency grid for the latter methods should be selected much smaller than $1/(t_N - t_1)$; more concretely we found the following choice of Δ to work well for a wide range of cases:

$$\Delta = \frac{1}{c(t_N - t_1)} \quad (8.11)$$

where $c \geq 3$ (we will use $c = 10$ in the numerical examples). Once f_{max} and Δ are chosen (as above), K follows immediately from (8.4).

8.2.2 Theoretical guideline

A spectral line can be located anywhere in the generic frequency bin $I_k = [\mu_k, \mu_{k+1}]$. Consequently, in principle Δ (or K) should be chosen such that the vectors $a(f)$, for $f \in I_k$, are (nearly) parallel to $a(v_k)$ (which is the column of A corresponding to the bin under discussion). Mathematically, this requirement can be formulated as :

$$\text{rank}(\Gamma_k) \approx 1 \quad (8.12)$$

where

$$\Gamma_k = \frac{\int_{\mu_k}^{\mu_{k+1}} a(f) a^*(f) df}{\mu_k}, \quad k = 1, \dots, K. \quad (8.13)$$

Let $\text{sinc}(x) = \sin(x)/x$. A simple calculation shows that:

$$\Gamma_k = \frac{1}{\Delta/2} \int_{-\Delta/2}^{\Delta/2} a(f + v_k) a^*(f + v_k) df = D_k C D_k^* \quad (8.14)$$

where

$$D_k = \begin{bmatrix} e^{i2\pi v_k t_1} & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & e^{i2\pi v_k t_N} \end{bmatrix} \quad (8.15)$$

and

$$\begin{aligned} C &= \frac{1}{\Delta/2} \int_{-\Delta/2}^{\Delta/2} a(f) a^*(f) df = \left\{ \int_{-\Delta/2}^{\Delta/2} e^{i2\pi f(t_k - t_p)} df \right\}_{k,p=1}^N \\ &= \{\Delta \text{sinc}[\pi \Delta(t_k - t_p)]\}_{k,p=1}^N. \end{aligned} \quad (8.16)$$

It follows from (8.14) that (8.12) can be reformulated in a more convenient k -independent form:

$$\text{rank}(C) \approx 1 \quad (8.17)$$

(because the modulation matrix D_k is nonsingular).

Theoretically, the above rank condition looks simple but its practical use is not without complication for at least two reasons :

i) Rank determination is numerically a delicate problem. One can determine the “practical rank” (prank) of C as :

$$\text{prank}(C) = \max_{k \in [1, \dots, N]} k \text{ s.t. } \lambda_k(C) > \rho \lambda_1(C) \quad (8.18)$$

where $\lambda_1(C) \geq \lambda_2(C) \geq \dots \geq \lambda_N(C)$ are the ordered eigenvalues of C , and $\rho < 1$ is a threshold. However the value of $\text{prank}(C)$ determined in this way is rather sensitive to the choice of ρ ; and the idea to choose a very small ρ (e.g. as in the rank determination function of Matlab) is not appealing for reasons exposed in ii) below.

ii) To achieve $\text{prank}(C) = 1$, for a very small ρ , typically a very large value of K is required. The rank condition then guarantees that the errors in modeling (8.2) via (8.8) are kept small. However the large number of parameters, K , that must be estimated in such a case inevitably leads to a large variance (and a high computational complexity).

It follows from the above discussion that the estimation variance is traded-off for bias as ρ in (8.17) and (8.18) decreases. While attempting to control

the variance-bias trade-off is at the heart of any biased estimation approach (such as the sparse estimation methods) there is no clear-cut rule for doing that and thus for choosing ρ in (8.18). With this fact in mind we will not pursue the idea of choosing K from (8.17) and (8.18), in order to avoid introducing an additional user parameter, viz. ρ (as too often done in some of the previous literature where the problem of selecting a certain user parameter is “resolved” by introducing a second parameter that is not easier to choose than the first one). Nevertheless, we will make use of (8.18) with $\rho = 10^{-3}$ to check at least roughly whether the sparse model in (8.8) corresponding to a given K is appropriate (which cannot be the case if, let us say, $\text{prank}(C) \geq 3$). Furthermore, the results in (8.14)-(8.16) above will turn out to be useful to the refined analysis in Section 8.3.

8.2.3 Numerical example

Consider the spectral line model in (8.2) with the following specifications :

Number of spectral lines : $n = 3$

Parameters : $f_1 = 0.4230, f_2 = 0.6875, f_3 = f_2 + \delta_f$

$\alpha_1 = 5, \alpha_2 = 5, \alpha_3 = 10$

Number of observations : $N = 50$

Sampling times : $t_1 = 0, t_N = 50$ [sec] and $\{t_k\}_{k=2}^{49}$ uniformly randomly distributed in the interval (0,50).

Noise : ε – normally distributed with zero mean and covariance matrix equal to $\sigma^2 I$.

The signal-to-noise ratio for this instance of (8.2) is defined as:

$$\text{SNR} = 10 \log_{10} [(\alpha_1^2 + \alpha_2^2 + \alpha_3^2)/\sigma^2] = 10 \log_{10}(150/\sigma^2) \quad (8.19)$$

Because $\min_k (t_{k+1} - t_k) < 0.5$ [sec] in this case, we have that $\tilde{f}_{max} > 1$ [Hz] and therefore we can set $f_{max} = 1$ [Hz] (assuming, for convenience, that we know no frequency in (8.2) is larger than 1 Hz). From (8.11) we get $\Delta = 2 \times 10^{-3}$ [Hz]. The practical guideline in sub-section 8.2.1 then suggests using $K = 500$.

For the frequency separation, δ_f , between f_2 and f_3 we will consider the following two values: $\delta_f = 0.01$ and 0.015 . The scenarios corresponding to these values can both be considered to be “high-resolution” estimation problems in the sense that the periodogram (whose resolution limit is $1/(t_N - t_1) = 0.02$ [Hz]) will likely be unable to resolve the two closely spaced spectral lines at f_2 and f_3 . Interestingly, all three sparse estimation methods considered in the next section work well for $\delta_f = 0.015$ Hz but some of them have difficulties in the case of $\delta_f = 0.01$ Hz unless K is properly chosen (see the next section for details).

In this section we will use the numerical example introduced above to show that :

- a) Sparse estimation methods are typically rather sensitive to the choice of K .
- b) The practical value of $K = 500$ recommended above can be shown to also be theoretically acceptable in the sense of the discussion in sub-section 8.2.2.

Figure 8.1 shows the spectra estimated by LIKES with $K = 300, 500$ and 2000 (see [92] for a detailed description of this method) in one realization with $\text{SNR} = 10\text{dB}$ and $\delta_f = 0.01$. As one can see from this figure, for $K = 500$ the estimated spectrum has three dominant spectral lines at locations close to the true frequencies $\{f_p\}_{p=1}^3$ and, in general terms, has a reasonable shape (although the heights of the spectral lines are slightly underestimated). For $K = 300$ and $K = 2000$, on the other hand, the estimated spectra look rather different from the true one : in the case of $K = 300$ the estimated spectrum has several large spurious peaks, whereas for $K = 2000$ the heights of the spectral line estimates are heavily biased downwards. This behavior can be explained as follows :

- i) For $K = 300$ the frequency bin is relatively large and this can lead to a situation in which a true spectral line vector, located in a generic bin I_k , is less correlated (in the sense of having a smaller scalar product) with vectors $a(f)$ for $f \in I_k$ than with vectors $a(f)$ whose f is far away from I_k ; the consequence of this being that the spectral power can leak into distant frequency cells.
- ii) For $K = 2000$ the frequency bin turns out to be quite small and therefore a true spectral line vector, associated with I_k , and vectors $a(f)$ corresponding to bins adjacent to I_k are nearly linearly dependent, the result being that the spectral power spreads over adjacent cells.

Remark 7 *Despite the fact that the heights of the three dominant peaks in Figure 8.1c are grossly underestimated, the locations of those peaks are close to the true frequencies. Therefore the frequency estimates obtained with LIKES for $K = 2000$ might have a reasonable accuracy, possibly comparable to that for $K = 500$. However the computational burden of LIKES is proportional to K and it is thus much larger at $K = 2000$ than at $K = 500$. The heavily biased amplitude estimates (due to an unnecessarily small frequency cell) and the higher computational burden associated with $K = 2000$ are evidently undesirable features.*

Figure 8.2 shows how $\text{prank}(C)$, determined using (8.18) with $\rho = 10^{-3}$, varies with $K \in [50, 2000]$. Observe that $\text{prank}(C) = 1$ for values of K from just above 500 up to 2000. This fact suggests that $K = 2000$ is indeed too large a value to use (as it is much larger than the smallest value of K that leads to $\text{prank}(C) = 1$), and also that $K = 500$ and perhaps $K = 300$ as well (at which $\text{prank}(C) = 2$) are reasonable choices. From Figure 8.1, however, we know that in fact $K = 300$ is much worse a choice than $K = 500$. In the next section we will present a simple method that can be used to decide between these two values of K (or any other values for that matter). Note that the minimum

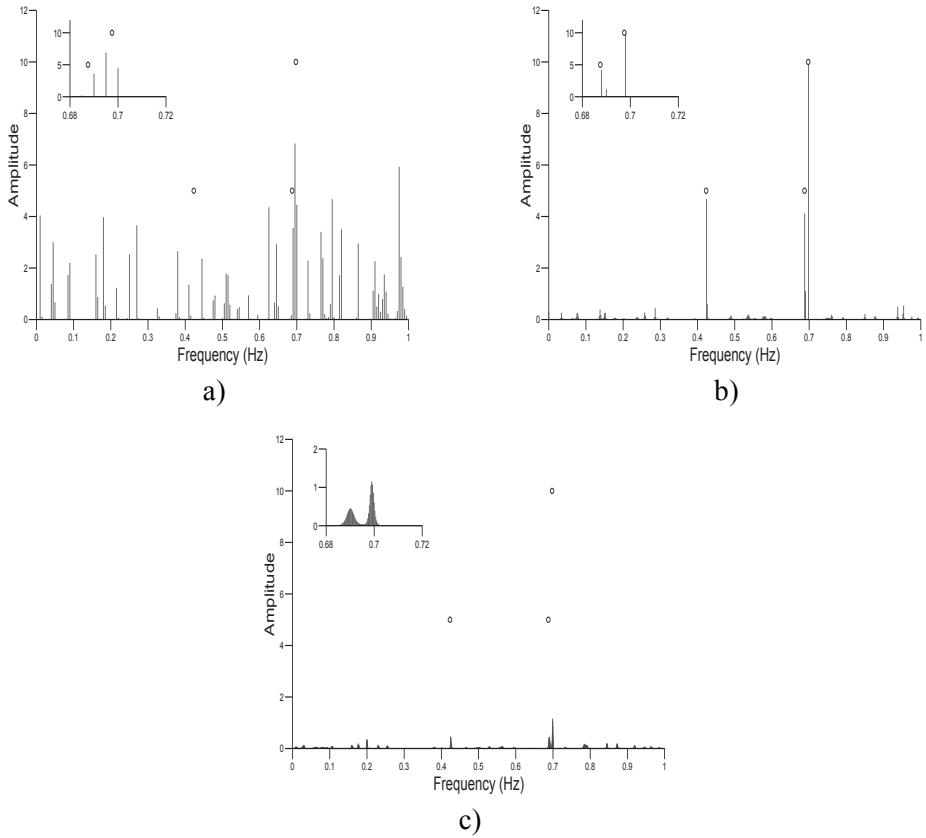


Figure 8.1. Estimated spectra for one realization with SNR = 10dB and $\delta_f = 0.01$: a) $K = 300$, b) $K = 500$, and c) $K = 2000$ (the true spectral lines are indicated by small circles). The inserts show the details of the estimated spectra for the frequency interval [0.68, 0.72] [Hz]

possible value of K that can be chosen is also limited by the desired frequency resolution.

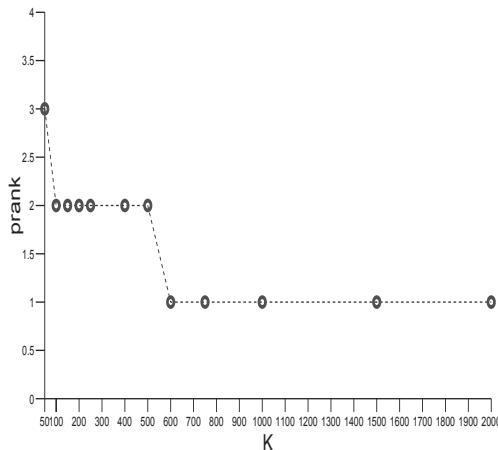


Figure 8.2. The variation of $\text{prank}(C)$, determined using (8.18) with $\rho = 10^{-3}$, for $K \in [50, 2000]$

8.3 Grid selection : refined ideas

8.3.1 LIKES

LIKES estimates β from the data model in equation (8.8) (which is repeated here for reader's convenience)

$$y = A\beta + \varepsilon \quad (8.20)$$

using the apriori information that

$$\beta \text{ is a sparse vector} \quad (8.21)$$

(see [92] for details). Let $\hat{\beta}$ denote the so-obtained estimate of β . The locations of the dominant peaks of the estimated spectrum, $\{|\hat{\beta}_k|\}$, provide the frequency estimates and the corresponding $\{\hat{\beta}_k\}$ can be taken as the amplitude estimates. As is typical for many other sparse estimation methods, the LIKES frequency estimates are accurate but the accuracy of the amplitude estimates may leave something to be desired. Usually more accurate amplitude estimates can be simply obtained as the solutions to the following Least-Squares (LS) problem:

$$\min_{\alpha} \|y - [a(\hat{f}_1) \cdots a(\hat{f}_n)] \alpha\|^2 \quad (8.22)$$

where $\{\hat{f}_p\}$ are the LIKES frequency estimates and \hat{n} is the number of peaks of the estimated spectrum that are deemed to be significant. Let $\hat{\alpha}$ denote the estimated amplitude vector so obtained, and let

$$\text{LS}(K, \text{LIKES}) = \|y - [a(\hat{f}_1) \cdots a(\hat{f}_{\hat{n}})] \hat{\alpha}\|^2 \quad (8.23)$$

where the dependence of the right-hand side on K has been emphasized via notation. *The value of K can be chosen by minimizing the above LS criterion over a set of given values.* A good value of \hat{n} can be determined in the same way after adding a penalty term to (8.23) that increases with increasing \hat{n} (as in AIC or BIC); However in order to focus on the choice of K we simply assume in the numerical examples of the next section that $\hat{n} = n$ (we remind the reader that apriori knowledge of n is *not* needed to apply LIKES; an estimate of n is useful only aposteriori to infer the number of significant spectral lines present in the data). We will use the acronym LIKES(K_0) to denote the LIKES method with K determined as outlined above.

8.3.2 ReLIKES

In the discussion on Figure 8.2 we made the point that “too large” values of K can usually be dismissed by studying $\text{prank}(C)$. However values of K that might in fact be “too small” may still be retained for consideration. As we saw from Figure 8.1, the performance of LIKES (or, in fact, of virtually any other sparse estimation method) for “too small” a value of K is far from satisfactory. The basic reason for this is that the model in (8.8) is a poor approximation in such a case : indeed this model assumes that a single vector, viz. $a(v_k)$, is a good linear basis for all vectors $a(f)$ with $f \in I_k$, which cannot be true when the bin I_k is “too large”. Interestingly this situation can be relatively simply corrected by considering a more detailed model, as explained below.

Let $\text{prank}(C) = r$ and let $U \in \mathbb{C}^{N \times r}$ be the matrix made from the r dominant eigenvectors of C . Then

$$B_k = D_k U \quad (8.24)$$

with D_k as defined in (8.15), can be expected to be a good linear basis for the vectors $a(f)$ corresponding to the k -th frequency bin I_k . In other words, for any $f \in I_k$ and $\alpha \in \mathbb{C}$ there is a vector $\gamma_k \in \mathbb{C}^{r \times 1}$ such that (with a good approximation) :

$$\alpha a(f) = B_k \gamma_k \quad k = 1, \dots, K \quad (8.25)$$

Consequently, in the case of $r > 1$, the following is a more accurate data representation than (8.8) :

$$y = B\gamma + \varepsilon; B = [B_1 \cdots, B_K]; \gamma = [\gamma_1^T \cdots \gamma_K^T]^T \quad (8.26)$$

where

$$[\|\gamma_1\| \cdots \|\gamma_K\|]^T = \text{is a sparse vector} \quad (8.27)$$

We designate the Refined LIKES method that estimates γ from (8.26) using the apriori information in (8.27) by the acronym ReLIKES (see [92] for a *general* description of such an extended LIKES algorithm). Let $\{\hat{\gamma}_k\}_{k=1}^K$ denote the so-obtained estimates of $\{\gamma_k\}$, and let p denote an index at which the sequence $\{\|\gamma_1\| \cdots \|\gamma_K\|\}$ has a dominant peak. For each such dominant peak we estimate the amplitude and frequency of an associated spectral line as the solutions to the problem :

$$\min_{\alpha, f \in I_p} \|\alpha a(f) - B_p \hat{\gamma}_p\|^2 \quad (8.28)$$

As is well known, the minimizing α is given by

$$\hat{\alpha} = a^*(\hat{f}) B_p \hat{\gamma}_p / N \quad (8.29)$$

where \hat{f} is obtained from:

$$\max_{f \in I_p} |a^*(f) B_p \hat{\gamma}_p|^2 \quad (8.30)$$

The frequency cell I_p has a small extent and thus the 1D maximization problem above can be readily solved, e.g. via a grid search.

Similarly to LIKES, once the ReLIKES frequency estimates are derived as explained above, we can obtain improved amplitude estimates as in (8.22); furthermore, we can select the value of K for ReLIKES by minimizing $\text{LS}(K, \text{ReLIKES})$ (see (8.23)) over a given range of values, once again as we did for LIKES. The ReLIKES method with K determined in this way is designated by the acronym ReLIKES(K_0).

8.3.3 SeLIKES

As r in ReLIKES increases we trade-off variance for bias. In other words, ReLIKES with $r \geq 2$ is likely to yield an estimate that has less bias but larger variance than the LIKES estimate using the same K . In general, therefore, there is no guarantee that the mean-square error (MSE) of ReLIKES is smaller than that of LIKES. With this fact in mind, we introduce a third method called Selected LIKES (SeLIKES) :

$$\text{SeLIKES} = \begin{cases} \text{LIKES} & \text{if } \text{LS}(K, \text{LIKES}) \leq \text{LS}(K, \text{ReLIKES}) \\ \text{ReLIKES} & \text{otherwise} \end{cases} \quad (8.31)$$

The particular instance of SeLIKES, for which K is selected as was done above for LIKES and ReLIKES, is designated by the abbreviation SeLIKES(K_0). Note that the SeLIKES estimates are directly obtained from the LIKES or ReLIKES estimate essentially without any additional computation.

Owing to the high dimensionality of the data models in (8.8) and in (8.26) it is difficult to make definite claims about the relative performance of the sparse estimation methods under consideration. In principle, we can expect ReLIKES to outperform LIKES in high-SNR regimes where the estimation bias dominates the variance, and vice-versa for low-SNR cases. Furthermore SeLIKES is expected to be better (and sometimes much better) than both LIKES and ReLIKES, for a given K , and *SeLIKES(K_0) to be the best method of those considered*. More concretely, in scenarios of small or medium difficulty all three methods can be expected to perform well, with SeLIKES(K_0) still being the best but not by much. On the other hand, in difficult scenarios (such as the one with $\delta_f = 0.01$ introduced above), SeLIKES(K_0) can outperform the other methods significantly. Indeed in such difficult cases outlying estimates can be relatively frequent (e.g., LIKES or ReLIKES can pick up erroneous spectral peaks, which leads to gross estimation errors). However, such outlying estimates might be produced by either LIKES or ReLIKES but not by both, and they might not occur for all values of K considered; consequently, the selection done in SeLIKES(K_0) between LIKES and ReLIKES and between different values of K can be expected to reduce the number of outlying estimates that would otherwise inflate the MSE considerably. The previous claims will be numerically illustrated, at least partly, in the next section.

8.4 Numerical illustrations and concluding remarks

The data were generated as described in Section 8.2, with $\delta_f = 0.01$ or 0.015 and with the SNR in the range $[0\text{dB}, 40\text{dB}]$. The estimation methods, viz. LIKES, ReLIKES and SeLIKES, were used with the following three values of K : 300 , 500 or K_0 ; we use $r = 2$ in ReLIKES for both $K = 300$ and $K = 500$, see Figure 8.2, and we select K_0 as the best value of K out of the two values mentioned above (i.e. 300 and 500). Figure 8.3 shows the average MSE of the frequency estimates obtained with these methods, as inferred from 100 Monte-Carlo simulations (average MSE = $[\text{MSE}(\hat{f}_1) + \text{MSE}(\hat{f}_2) + \text{MSE}(\hat{f}_3)]/3$).

The results presented in the figure lend support to the claims made at the end of the previous section. In particular, we can see that SeLIKES(K_0) is indeed the best method in all cases considered and therefore that *SeLIKES(K_0) should be the method of choice*. Note that the need to run ReLIKES, in addition to LIKES, to compute the SeLIKES estimate more than doubles the computational burden. If this is a concern then using LIKES(K_0) or preferably ReLIKES(K_0) in lieu of SeLIKES(K_0) may suffice, especially if the scenario dealt with is not a “difficult” one. Finally, we remark on the fact that the ideas underlying both ReLIKES and the grid selection scheme proposed here can be applied verbatim to other sparse estimation methods for which they are likely to work as well as they did for LIKES.

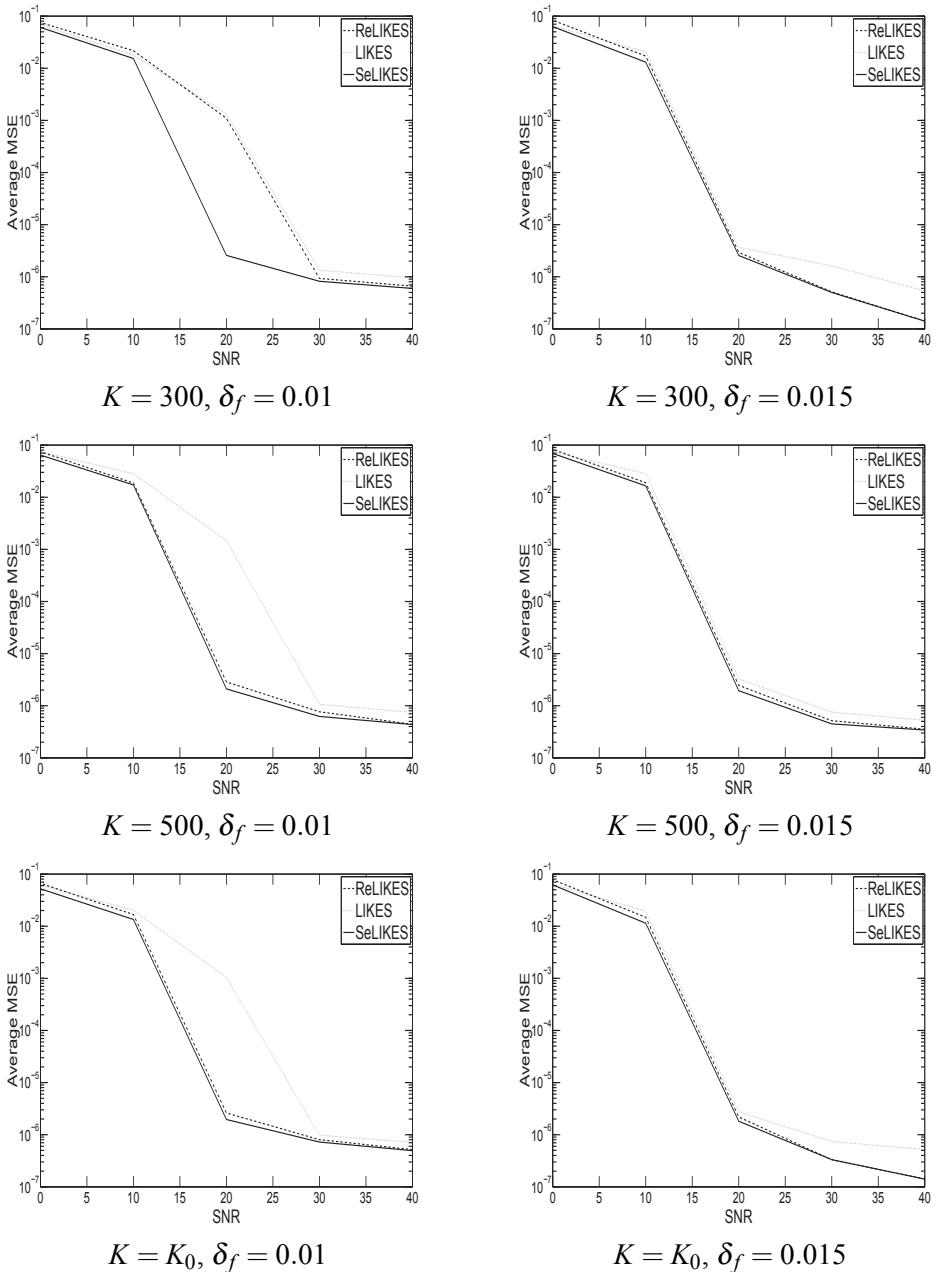


Figure 8.3. Average MSE vs SNR for the frequency estimates obtained with LIKES, ReLIKES and SeLIKES.

9. Sparse spectral-line estimation for non-uniformly sampled multivariate time series

9.1 Data model and problem formulation

Let $\{y(t) \stackrel{\Delta}{=} [y_1(t), \dots, y_M(t)]^T \in \mathbb{C}^{M \times 1}\}$ denote an M -variate time series with each of its components satisfying the following spectral-line model:

$$y_m(t_n) = \sum_{l=1}^{C_m} r_{l,m} e^{i\Omega_{l,m} t_n} + e_m(t_n) \quad n = 1, \dots, N \quad m = 1, \dots, M \quad (9.1)$$

where $\{t_n\}_{n=1}^N$ denote the sampling times which can be nonuniformly placed. For any m , $\{r_{l,m} \in \mathbb{C}\}_{l=1}^{C_m}$ denote the amplitudes of the C_m components located at the frequencies $\{\Omega_{l,m} \in [0, \Omega_{max}], \Omega_{max} \in \mathbb{R}\}_{l=1}^{C_m}$, respectively, and $\{e_m(t_n)\}$ denotes the noise in the data. Given $\{y_m(t_n)\}_{m=1,n=1}^{M,N}$ we want to estimate, for $m = 1, \dots, M$, the number of components C_m and their corresponding amplitudes $\{r_{l,m}\}_{l=1}^{C_m}$ and frequencies $\{\Omega_{l,m}\}_{l=1}^{C_m}$. Thus the problem is to *detect* the components in the model as well as to *estimate* their amplitudes and frequencies, see e.g. [98] [22] [47] and [110] for motivation of this problem and some possible applications.

To tackle this problem, we divide the frequency interval $[0, \Omega_{max}]$ into a set of uniformly spaced values with a spacing of Δ and form a fine grid $\{\omega_l\}_{l=1}^K$ such that the true frequencies lie on (or, practically, close to) the grid i.e., $\{\Omega_{l,m}\}_{l=1,m=1}^{C_m,M} \subset \{\omega_l\}_{l=1}^K$. A typical choice of Δ is $\frac{2\pi}{10(t_N - t_1)}$ [98] [93]. Making use of the grid, the data model in (9.1) can be re-written as follows :

$$y_m \stackrel{\Delta}{=} \begin{bmatrix} y_m(t_1) \\ \vdots \\ y_m(t_N) \end{bmatrix} = \begin{bmatrix} e^{i\omega_1 t_1} & \dots & e^{i\omega_K t_1} \\ \vdots & \dots & \vdots \\ e^{i\omega_1 t_N} & \dots & e^{i\omega_K t_N} \end{bmatrix} \begin{bmatrix} x_{1,m} \\ \vdots \\ x_{K,m} \end{bmatrix} + \begin{bmatrix} e_m(t_1) \\ \vdots \\ e_m(t_N) \end{bmatrix} \quad m = 1, \dots, M \quad (9.2)$$

for some $\{x_{l,m}\}$ (see below). Let

$$\begin{aligned} Y &\stackrel{\Delta}{=} [y_1 \quad \cdots \quad y_M], X \stackrel{\Delta}{=} \begin{bmatrix} x_{1,1} & \cdots & x_{1,M} \\ \vdots & \cdots & \vdots \\ x_{K,1} & \cdots & x_{K,M} \\ e^{i\omega_1 t_1} & \cdots & e^{i\omega_K t_1} \end{bmatrix} \\ A &\stackrel{\Delta}{=} [a_1 \quad \cdots \quad a_K] = \begin{bmatrix} \vdots & \cdots & \vdots \\ e^{i\omega_1 t_N} & \cdots & e^{i\omega_K t_N} \end{bmatrix} \\ E &\stackrel{\Delta}{=} \begin{bmatrix} e_1(t_1) & \cdots & e_M(t_1) \\ \vdots & \cdots & \vdots \\ e_1(t_N) & \cdots & e_M(t_N) \end{bmatrix}. \end{aligned} \quad (9.3)$$

We will refer to $\{y_m\}$ as the data snapshots. By using the above matrix notation, (9.2) can be re-written as follows :

$$\begin{aligned} Y &= AX + E \\ &= [A \ I] \begin{bmatrix} X \\ E \end{bmatrix} \stackrel{\Delta}{=} BZ \end{aligned} \quad (9.4)$$

where I denotes the identity matrix of dimension $N \times N$, and $B \stackrel{\Delta}{=} [b_1, \dots, b_{K+N}] = [AI]$. Usually, the number of components $\{C_m\}$ is much smaller than the grid dimension K , so that only a few values of $\{x_{l,m}\}$ are non-zero i.e., for any m there exist a set $\{m_l\}$ such that $\{x_{m_l,m} = r_{l,m}\}$ or, in other words, the matrix X is row-sparse. Thus, we have transformed the nonlinear detection and estimation problem associated with (9.1) into a linear sparse parameter estimation problem for (9.4).

By assuming that the noise sequences in the data snapshots have zero means but possibly different variances, namely $\{\sigma_k\}_{k=1}^N$, and that they are uncorrelated with each other as well as with X , the (normalized) covariance matrix of Y can be expressed as :

$$R \stackrel{\Delta}{=} E[YY^*]/M = \sum_{l=1}^K \sum_{m=1}^M \frac{E[|x_{l,m}|^2]}{M} a_l a_l^* + \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_N \end{bmatrix} \stackrel{\Delta}{=} BPB^* \quad (9.5)$$

where

$$P = \begin{bmatrix} \sum_{m=1}^M \frac{E[|x_{1,m}|^2]}{M} & 0 & \dots & \dots & \dots & 0 \\ 0 & \sum_{m=1}^M \frac{E[|x_{2,m}|^2]}{M} & 0 & \dots & \dots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \sigma_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \sigma_N \end{bmatrix} \quad (9.6)$$

$$\Delta \triangleq \begin{bmatrix} p_1 & 0 & \dots & \dots & \dots & 0 \\ 0 & p_2 & 0 & \dots & \dots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & p_{K+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & p_{K+N} \end{bmatrix}$$

where $E(\cdot)$ denotes the expectation operation. The data sample covariance matrix can be obtained as :

$$\hat{R} = \frac{1}{M} \sum_{m=1}^M y_m y_m^* = YY^*/M. \quad (9.7)$$

The quantities $\{p_l\}_{l=1}^K$ in R represent the powers at the frequencies $\{\omega_l\}_{l=1}^K$, respectively. Although our primary interest is in the estimation of X from Y , we will also estimate $\{p_l\}$. In fact, in the methods described in the following section we will start with the problem of estimating $\{p_l\}$ and an estimate of X will be obtained as a byproduct of the method. We would also like to point out that apart from estimating the powers at different frequencies, we also estimate the noise variances by estimating the quantities $\{p_l\}_{l=K+1}^{K+N}$.

In Section 9.2, we introduce two new methods, namely SPICE and LIKES, for estimating $\{p_l\}$ and X , briefly describe the multivariate-SBL (MSBL) algorithm and duly refer to the previous relevant literature. In Section 9.3, we compare the statistical performance of SPICE, LIKES and MSBL, as well as their computational complexities and convergence properties, by means of numerical simulations.

9.2 SPICE, LIKES and MSBL

9.2.1 SPICE

Given \hat{R} , the $\{p_k\}$ in R can be estimated as the solutions to the following minimization problem :

$$\min_p \|R^{-1/2}(R - \hat{R})\|^2 \quad (9.8)$$

where $\|\cdot\|$ denotes the Frobenius matrix norm, $R^{-1/2}$ denotes a Hermitian square root of R^{-1} , and $p \triangleq [p_1, \dots, p_{K+N}]^T$ with each $p_k \geq 0$. We have used this type of covariance fitting criterion in [93] (for the spectral analysis of univariate time series, i.e. $M = 1$) and, in a related form, in [34] (for spatial spectral analysis, which is essentially equivalent to multivariate time series analysis with $M \geq N$) to derive a sparse parameter estimation technique named **SPICE** (**s**parse **i**terative **c**ovariance **b**ased **e**stimation). Here we extend SPICE to the multivariate case with $M \in (1, N)$. By substituting the expression for R in (9.8) and expanding the cost function we get the following equivalent formulation of the problem:

$$\min_p \text{tr}(\hat{R}^* R^{-1} \hat{R}) + \text{tr}(R) \quad (9.9)$$

or

$$\min_p \text{tr}(\hat{R}^* R^{-1} \hat{R}) + \sum_{k=1}^{K+N} w_k^2 p_k \quad (9.10)$$

where $w_k = \|b_k\|$ and $\text{tr}(\cdot)$ denotes the matrix trace. The minimization problem in (9.10) is convex and has a unique global minimum. In fact, (9.10) can be cast as the following semi-definite program (SDP),

$$\begin{aligned} & \min_{p, \{\alpha_l\}} \sum_{l=1}^N \alpha_l + \sum_{k=1}^{K+N} w_k^2 p_k \\ & \text{s.t. } \begin{bmatrix} \alpha_l & g_l^* \\ g_l & R \end{bmatrix} \geq 0 \quad l = 1, \dots, N \end{aligned} \quad (9.11)$$

where $\{\alpha_l\}$ are auxiliary variables and $[g_1, \dots, g_N] = \hat{R}$. However, solving the SDP in (9.11) can be quite time consuming: for instance this SDP cannot be solved on a general purpose PC even for relatively modest dimensions (say $N = 100$, $M = 10$ and $K = 1000$). To tackle this computational issue, we follow [93] to derive an iterative algorithm for the problem in (9.10).

Consider the following augmented problem:

$$\begin{aligned} & \min_{p, Q} \text{tr}(Q^* P^{-1} Q) + \sum_{k=1}^{K+N} w_k^2 p_k \\ & \text{s.t. } BQ = \hat{R}. \end{aligned} \quad (9.12)$$

Minimization over Q (for fixed p) is straightforward: the solution is given by $Q_0 = PB^* R^{-1} \hat{R}$ (see, e.g., [98], Appendix A, Result R35 and [34], Section III,

Page 632). It is easy to verify that substituting Q_0 back into the cost function in (9.12) yields the original problem in (9.10). Hence the p 's obtained from (9.10) and (9.12) must be identical.

The minimization over p (for a given Q) can also be done analytically as follows. Using $Q = [\beta_1, \dots, \beta_{K+N}]^*$, the optimization problem in (9.12) (for fixed $\{\beta_k\}$) can be reduced to

$$\min_p \sum_{k=1}^{K+N} \frac{\|\beta_k\|^2}{p_k} + \sum_{k=1}^{K+N} w_k^2 p_k. \quad (9.13)$$

A simple calculation shows that

$$\begin{aligned} \sum_{k=1}^{K+N} \left(\frac{\|\beta_k\|}{\sqrt{p_k}} - w_k \sqrt{p_k} \right)^2 &\geq 0 \iff \\ \sum_{k=1}^{K+N} \left(\frac{\|\beta_k\|^2}{p_k} + w_k^2 p_k - 2w_k \|\beta_k\| \right) &\geq 0 \iff \\ \sum_{k=1}^{K+N} \left(\frac{\|\beta_k\|^2}{p_k} + w_k^2 p_k \right) &\geq \sum_{k=1}^{K+N} 2w_k \|\beta_k\|. \end{aligned} \quad (9.14)$$

The left hand side in the above inequality is nothing but the cost function in (9.13) and the equality holds only when $p_k = \|\beta_k\|/w_k$. Thus the minimizer of (9.13) is

$$p_k = \frac{\|\beta_k\|}{w_k} \quad k = 1, \dots, K+N. \quad (9.15)$$

Since the cost function in (9.12) is convex in both Q and p , the cyclic minimization over Q and p , i.e. minimization over Q (for fixed p) and vice-versa, starting from any arbitrary initial point will lead to the global minimum of (9.12). The $(i+1)$ -th iteration of the so-obtained cyclic algorithm consists of the following steps:

$$\begin{aligned} Q^{i+1} &= P^i B^* R^{-1}(i) \hat{R} \\ p_k^{i+1} &= \frac{\|\beta_k^{i+1}\|}{w_k} \quad k = 1, \dots, K+N \\ R(i+1) &= B P^{i+1} B^*. \end{aligned} \quad (9.16)$$

(for initialization of (9.16)), as well as of the other yet-to-be derived algorithms, see Section 9.3) An estimate of Z , and hence of X can then be obtained as follows :

$$\begin{aligned} \hat{Z} &= Q_c Y (Y^* Y / M)^{-1} \\ \hat{X} &= \text{the first } K \text{ rows of } \hat{Z} \end{aligned} \quad (9.17)$$

where Q_c denotes the value of Q obtained at the convergence of (9.16).

9.2.2 LIKES

LIKES, which stands for **l**ikelihood-based **e**stimation of **s**parse parameters, estimates p by minimizing the Gaussian negative log-likelihood (NLL) function

$$\begin{aligned} f(p) &= \text{tr}(R^{-1}\hat{R}) + \ln|R| \\ &= \frac{1}{M}\text{tr}(Y^*R^{-1}Y) + \ln|R| \end{aligned} \quad (9.18)$$

that is

$$\min_p f(p). \quad (9.19)$$

The minimization problem in (9.19) is non-convex. In fact, it can be shown that the two terms in $f(p)$ are convex and concave in p , respectively. In [92] we have derived a LIKES iterative algorithm based on a majorization-minimization technique that minimizes the above cost function in the univariate case. Here, following a similar approach to that in [92], we extend the LIKES algorithm to the multivariate case.

Since the second term in $f(p)$, viz. $\ln|R|$, is a concave function in p , it can be majorized by its tangent plane at any point. Let \tilde{p} be an arbitrary point in the parameter space, and let \tilde{R} denote the corresponding covariance matrix; then :

$$\begin{aligned} \ln|R| &\leq \ln|\tilde{R}| + \sum_{k=1}^{K+N} \text{tr}(\tilde{R}^{-1}b_k b_k^*) (p_k - \tilde{p}_k) \\ &= \ln|\tilde{R}| - N + \sum_{k=1}^{K+N} \tilde{w}_k^2 p_k \end{aligned} \quad (9.20)$$

where

$$\tilde{w}_k^2 = b_k^* \tilde{R}^{-1} b_k. \quad (9.21)$$

It follows from (9.20) that :

$$f(p) \leq (\ln|\tilde{R}| - N) + \frac{1}{M}\text{tr}(Y^*R^{-1}Y) + \sum_{k=1}^{K+N} \tilde{w}_k^2 p_k \stackrel{\Delta}{=} g(p) \quad (9.22)$$

for any vectors \tilde{p} and p . Note also that

$$f(\tilde{p}) = g(\tilde{p}). \quad (9.23)$$

The main implication of (9.22) and (9.23) is that we can decrease the function $f(p)$ from $f(\tilde{p})$ to, let us say, $f(\hat{p})$ by choosing \hat{p} as a minimizer of the majorizing function $g(p)$ or at least such that $g(\tilde{p}) > g(\hat{p})$:

$$f(\hat{p}) \leq g(\hat{p}) < g(\tilde{p}) = f(\tilde{p}) \quad (9.24)$$

This is precisely the basic idea behind the majorization-minimization approach to solve the problem in (9.19), see, e.g. [101] and the references therein. The usefulness of this approach depends on how much easier the minimization (or the decrease) of $g(p)$ is compared to minimizing $f(p)$ directly. In the present case, minimizing $g(p)$ is much easier than minimizing $f(p)$ because $g(p)$ is (to within an additive constant) a SPICE-like convex criterion function (compare it with (9.10) after replacing \hat{R} by Y/\sqrt{M}). Consequently, the approach used

in the previous subsection can be adopted verbatim to find a vector \hat{p} with the above property, for any given \tilde{p} .

Following the said approach, we consider the augmented optimization problem :

$$\begin{aligned} \min_{p, \tilde{Q}} & \frac{1}{M} \text{tr}(\tilde{Q}^* P^{-1} \tilde{Q}) + \sum_{k=1}^{K+N} \tilde{w}_k^2 p_k \\ \text{s.t. } & B\tilde{Q} = Y \end{aligned} \quad (9.25)$$

The minimizer \tilde{Q} (for fixed p) is given by $\tilde{Q}_0 = PB^*R^{-1}Y$ and substituting \tilde{Q}_0 into (9.25) yields the cost function in (9.22) (to within an additive constant). By using $\tilde{Q} = [\tilde{\beta}_1, \dots, \tilde{\beta}_{K+N}]^*$ and a calculation similar to (9.13)-(9.14), the minimizer p (for fixed \tilde{Q}) can also be derived. The resulting LIKES algorithm comprises an inner loop to minimize (or to decrease) $g(p)$ and an outer loop to recompute the weights $\{\tilde{w}_k\}$, and can be summarized as follows:

The inner loop:

At iteration $i+1$:

$$\begin{aligned} \tilde{Q}^{i+1} &= P^i B^* R^{-1}(i) Y \\ p_k^{i+1} &= \frac{\|\tilde{\beta}_k^{i+1}\|}{\sqrt{M\tilde{w}_k}} \quad k = 1, \dots, K+N \\ R(i+1) &= BP^{i+1}B^* \end{aligned}$$

The outer loop:

Letting $R_c (\{p_k^c\})$ denote the $R (\{p_k\})$ obtained at convergence (or after a pre-specified number of iterations) of the inner loop, compute

$$\tilde{w}_k = \sqrt{b_k^* R_c^{-1} b_k}$$

and then go to the inner loop. The inner loop will then be initialized with $\{p_k^c\}$.

Final step :

The estimates of Z and X are obtained as

$$\begin{aligned} \hat{Z} &= \tilde{Q}_c \\ \hat{X} &= \text{the first } K \text{ rows of } \hat{Z} \\ \text{where } \tilde{Q}_c &\text{ denotes the value of } \tilde{Q} \text{ obtained at convergence of the outer loop.} \end{aligned} \quad (9.26)$$

9.2.3 MSBL

Besides the majorization-minimization technique, the NLL function in (9.18) can also be minimized by an expectation-maximization (EM) algorithm. Such an approach named **sparse Bayesian learning** (SBL) has been suggested in [118] for the univariate case; and in [119] for the multivariate case where it was called MSBL. For conciseness we present only the main steps of the

MSBL algorithm:

Iterative step :

$$\begin{aligned}\tilde{Q}^{i+1} &= P^i B^* R^{-1}(i) Y \\ p_k^{i+1} &= p_k^i - (p_k^i)^2 b_k^* R^{-1}(i) b_k + \|\tilde{\beta}_k^{i+1}\|^2 / M \quad k = 1, \dots, K+N \\ R(i+1) &= B P^{i+1} B^*.\end{aligned}$$

Final step :

The estimates of Z and X are obtained as:

$$\hat{Z} = \tilde{Q}_c$$

\hat{X} = the first K rows of \hat{Z}

where \tilde{Q}_c denotes the value of \tilde{Q} obtained at the convergence.

(9.27)

9.3 Numerical simulations and concluding remarks

In this section we numerically compare the performance of SPICE, LIKES and MSBL. The data were generated via the model in (9.2) with $N = 100$, $M = 10$ and $K = 1000$. The sampling times $\{t_n\}$ were uniformly randomly distributed between $[0 - 20]$ sec. The value of Ω_{max} was chosen to be 10π rad/sec. In each of the 10 data snapshots, 5 sinusoidal components were present. Each snapshot shares three common frequencies with its neighboring snapshots. Table 9.1 shows the values of the frequencies with non-zero amplitudes in different snapshots. The amplitudes of all existing sinusoidal components were chosen as 5. The noise was Gaussian distributed with zero mean and variance equal to σ . The signal to noise ratio (SNR) is defined by $10 \log(25/\sigma)$. The three methods were initialized with the periodogram estimate $\{p_k^0 = b_k^* \hat{R} b_k / N\}$. For all methods the convergence criterion used to terminate the iterations was : $\frac{\|p^{i+1} - p^i\|}{\|p^i\|} < 10^{-3}$. In the case of LIKES, this convergence criterion has been used in the outer loop while the inner loop has been run for 10 iterations.

9.3.1 Statistical performance

Figure 9.1 shows 100 superimposed plots of amplitude spectra corresponding to the (randomly picked) 7-th data snapshot obtained with SPICE, LIKES and MSBL. The spectra obtained with all methods are sparse and they correctly indicate the presence of sinusoidal components in most of the Monte-Carlo runs; furthermore the likelihood based approaches provide more accurate amplitude estimates than SPICE. Figure 9.2 shows the plots of average mean square error (AMSE) of the amplitude estimates at the true frequency locations, as well as the probability of detection of the true frequencies vs SNR. The AMSE of the

Frequencies $\times \pi$ (rad/sec)	Snapshot number									
	1	2	3	4	5	6	7	8	9	10
0.05	0.79	0.85	1.24	1.45	1.53	1.66	1.84	2.40	2.63	
0.76	0.84	1.07	1.37	1.46	1.63	1.82	2.29	2.60	2.64	
0.79	0.85	1.24	1.45	1.53	1.66	1.84	2.40	2.63	3.12	
0.84	1.07	1.37	1.46	1.63	1.82	2.29	2.60	2.64	3.51	
0.85	1.24	1.45	1.53	1.66	1.84	2.40	2.63	3.12	4.00	

Table 9.1. The frequencies of the sinusoidal components in the different data snapshots.

amplitude estimates was calculated as:

$$\text{AMSE} = \frac{1}{5000} \sum_{k=1}^{100} \sum_{m=1}^{10} \sum_{\tilde{l}=1}^5 \left| \hat{x}_{m\tilde{l},m}^k - 5 \right|^2 \quad (9.28)$$

where $\{m_l\}_{l=1}^5$ denote the frequency indices corresponding to the 5 sinusoidal components present in the m -th snapshot, and the superscript k in $\hat{x}_{m\tilde{l},m}^k$ denotes the Monte-Carlo run. The probability of detection was computed by first picking the 5 dominant peaks of the estimated spectrum, sorting the frequencies corresponding to those peaks, and then calculating the mean absolute error of those frequency estimates: if the mean absolute error is equal to zero then we have a detection, else we declare a miss. It can be observed from the plots in Figure 9.2 that SPICE is less accurate than LIKES and MSBL for both amplitude and frequency estimation, and that LIKES is slightly better than MSBL in the case of amplitude estimation.

9.3.2 Complexity and convergence rate

The computational complexity per iteration (in flops) of the considered methods is on the order of $O(2N^3 + 2N^2M + 2KMN + KN^2 + KM + NM)$ with MSBL requiring an additional $KN^2 + N^3$ flops to compute the terms in the power update formula, see (9.27). However, the convergence rates of the three methods differ quite a bit from one another, which leads to rather different execution times. Figure 9.3a shows the average computation times (in sec) per run vs SNR; it is seen from this figure that the times decrease with increasing SNR, which is primarily due to the fact that the methods converge faster as the SNR increases. For a fixed SNR, SPICE is faster than LIKES which is faster than MSBL.

As both LIKES and MSBL minimize the same NLL criterion, it is interesting to compare their convergence rates and the values of NLL they attain at convergence. Figure 9.3b shows the NLL value vs the iteration number for LIKES and MSBL in one Monte-Carlo run; it can be seen from this plot

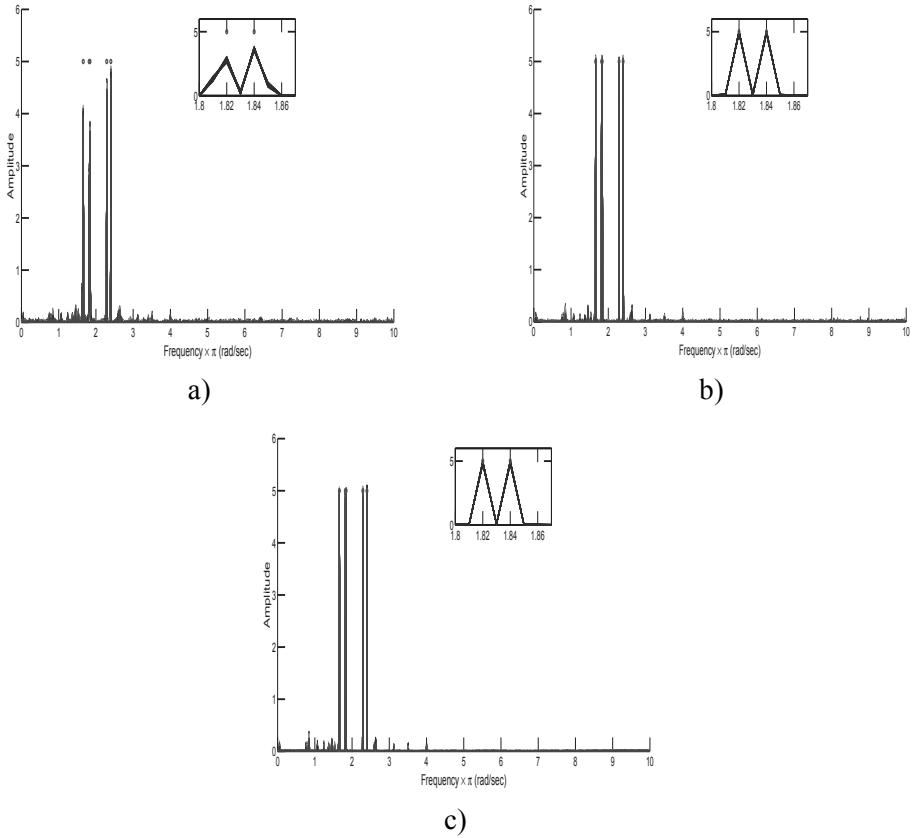


Figure 9.1. One hundred superimposed plots of the amplitude spectra corresponding to the 7-th data snapshot estimated via a) SPICE, b) LIKES and c) MSBL. The circles indicate the locations and amplitudes of the true components in the data. The SNR was 40 dB. The peaks at the closely-spaced frequencies of 1.82π and 1.84π appear almost merged but in actuality they are distinct. The zoom-in plots show the spectrum in the interval $[1.8 - 1.86] \times \pi$ rad/sec to confirm this fact.

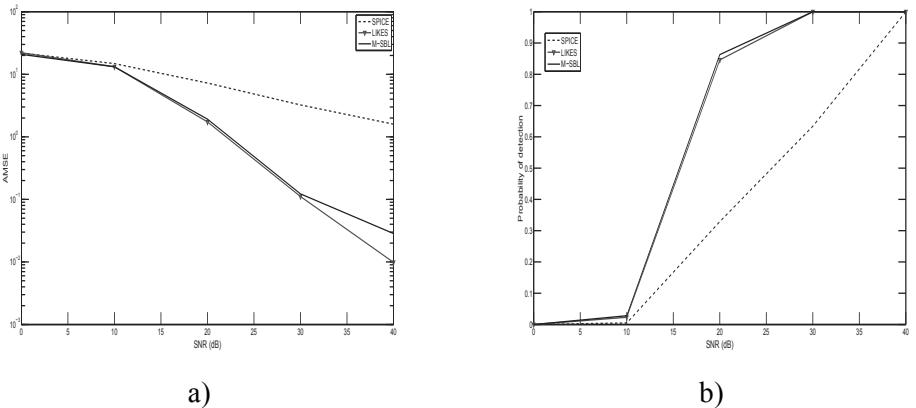
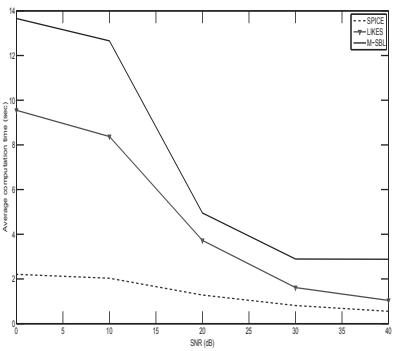


Figure 9.2. a) AMSE of amplitude estimates at the true frequency locations vs SNR
b) Probability of detection of true frequencies vs SNR. The number of Monte-Carlo runs was 100.

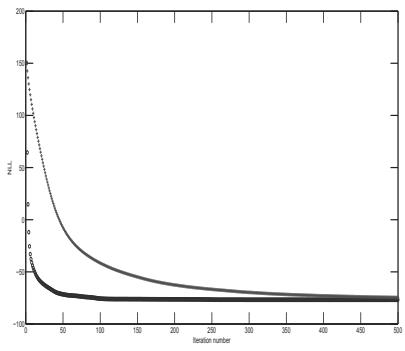
that LIKES converges faster than MSBL and also that the value of NLL that it attains at convergence is slightly lower than the NLL value attained by MSBL.

9.3.3 Concluding remarks

To conclude, we have introduced two methods to solve the problem of spectral-line analysis for nonuniformly sampled multivariate time series. Both methods yield sparse parameter estimates and they do not require any selection of hyperparameters. We have also considered the previously proposed method of MSBL that minimizes the same NLL function as LIKES. We compared these three methods via numerical simulations and observed that LIKES and MSBL are more accurate than SPICE but at the cost of extra computation time. Regarding LIKES and MSBL, we showed that LIKES converges faster than MSBL and also that the NLL value that LIKES attains at convergence can be slightly smaller than the NLL value attained by MSBL.



a)



b)

Figure 9.3. a) Average computation times (sec) of SPICE, LIKES and MSBL vs SNR,
b) NLL values of LIKES and MSBL vs iteration number (o : LIKES, + : MSBL).

Part III: Applications

10. Exoplanet detection : Analysis of radial velocity data

10.1 Introduction

Extrasolar planet (or shortly exoplanet) detection is a fascinating and challenging area of research in the field of astrophysics. Till mid 2012, over 300 exoplanets have been discovered. Some of the techniques available in the astrophysics literature to detect exoplanets are astrometry, the radial velocity method, pulsar timing, the transit method and gravitational microlensing. Among these methods, the radial velocity analysis is the most commonly used technique, in which the Doppler shift in the spectral lines and hence the radial velocity of the parent star is measured. The spectrum of the measured Doppler shifts is then analyzed to detect the exoplanet(s) revolving around the star.

Most often the radial velocity measurements are obtained at nonuniformly spaced time intervals due to hardware and practical constraints, which limits the application of commonly used spectral analysis methods. The most straightforward way to deal with this problem is to use the standard periodogram by ignoring the nonuniformity of data samples, which results in an inaccurate spectrum. In [82], a method named CLEAN was proposed, which is based on iterative deconvolution in the frequency domain to obtain a clean spectrum from an initial dirty one. A periodogram related method is the least squares periodogram (also called the Lomb-Scargle periodogram) [86] which estimates the sinusoidal components by fitting them to the observed data. In [14], a method based on sparse signal representation was proposed, in which the radial velocity data set is fitted to an over-complete dictionary of complex sinusoids under the constraint that the spectrum is sparse. Most recently, [122] introduced a new method called the Iterative Adaptive Approach (IAA), which relies on solving an iterative weighted least squares problem.

In this chapter, we extend IAA to radial velocity data analysis. We also derive and discuss regularized and computationally efficient versions of IAA. We then refine and determine the statistical significance of the spectral peaks by using a relaxation maximum likelihood algorithm (RELAX) and a generalized likelihood ratio test (GLRT). Numerical experiments are carried out on simulated and real life radial velocity data sets.

In Section 2, we describe the model used in this chapter for the radial velocity data. Section 3 describes the different IAA-based methods for estimating the spectrum of radial velocity data. Section 4 presents the RELAX and GLRT methods for refining and establishing the significance levels of the peaks of

spectrum, and Section 5 contains the numerical examples for real life radial velocity data sets. Finally, the chapter is concluded in Section 6.

10.2 Data Model

Let $\{y(t_n)\}_{n=1}^N$ denote the radial velocity of a star measured at a set of possibly nonuniform time instants $\{t_n\}_{n=1}^N$. Based on the Keplerian model of planetary motion [126] [26] [74], the radial velocity data are modeled as follows:

$$y(t_n) = C_0 + \sum_{m=1}^M \beta_m [\cos(\omega_m + v_m(t_n)) + e_m \cos(\omega_m)], \quad n = 1, \dots, N \quad (10.1)$$

where C_0 is the constant radial velocity, and

$$\begin{aligned} \tan(v_m(t_n)) &= \sqrt{\frac{1+e_m}{1-e_m}} \tan(E_m(t_n)) \\ E_m(t_n) - e_m \sin(E_m(t_n)) &= \frac{2\pi(t_n - T_m)}{P_m}, \end{aligned} \quad (10.2)$$

M : Number of exoplanets revolving around the star.

e_m : Eccentricity of the orbit of the m^{th} planet.

ω_m : Longitude of the periastron for the m^{th} planet.

P_m : Orbital period of the m^{th} planet; P_m is related to orbital frequency f_m by $f_m = \frac{1}{P_m}$.

T_m : Periastron passage time of the m^{th} planet.

β_m : Radial velocity amplitude of the m^{th} planet.

$v_m(t_n), E_m(t_n)$: True and eccentric anomaly of the m^{th} planet, with t_n denoting their time dependence.

We can re-write (10.1) as

$$\begin{aligned} y(t_n) &= C_0 + \sum_{m=1}^M \beta_m e_m \cos(\omega_m) + \sum_{m=1}^M \beta_m \cos(\omega_m + v_m(t_n)) \\ &= C_1 + \frac{1}{2} \sum_{m=1}^M (\beta_m \exp(j\omega_m) \exp(jv_m(t_n)) + \beta_m \exp(-j\omega_m) \exp(-jv_m(t_n))) \\ &= \sum_{m=-M}^M \alpha_m \exp(jv_m(t_n)), \end{aligned} \quad (10.3)$$

where $\alpha_0 = C_1 = C_0 + \sum_{m=1}^M \beta_m e_m \cos(\omega_m)$, $\alpha_m = \alpha_{-m}^* = \frac{\beta_m}{2} \exp(j\omega_m)$, $v_0(t_n) = 0$ and $v_m(t_n) = -v_{-m}(t_n)$. With noise in the measurements (10.3) becomes

$$y(t_n) = \sum_{m=-M}^M \alpha_m \exp(jv_m(t_n)) + \varepsilon(t_n), \quad n = 1, \dots, N, \quad (10.4)$$

where we assume that $\varepsilon(t_n)$ is real-valued zero-mean white Gaussian noise with variance σ^2 . Although $\alpha_m = \alpha_{-m}^*$, we will not impose this constraint

when we estimate $\{\alpha_m\}$ for the sake of simplicity. This relaxation will typically lead only to a negligible degradation in performance. We also note that since we model the real-valued radial velocity data with a complex-valued model, for implementation simplicity, we need both negative and positive frequency in the data model to represent a real-valued component.

The model discussed above can be expressed more compactly using the following notation:

$$\begin{aligned}\mathbf{y} &= [y(t_1), \dots, y(t_N)]', \\ \mathbf{s} &= [\alpha_{-M}, \dots, \alpha_M]', \\ \boldsymbol{\varepsilon} &= [\varepsilon(t_1), \dots, \varepsilon(t_N)]', \\ \mathbf{a}_m &= [\exp(jv_m(t_1)), \dots, \exp(jv_m(t_N))]'\end{aligned}\tag{10.5}$$

where $/$ denotes the transpose operation. With this notation the model in (10.4) can be rewritten as follows:

$$\mathbf{y} = \mathbf{As} + \boldsymbol{\varepsilon}\tag{10.6}$$

where $\mathbf{A} = [\mathbf{a}_{-M}, \dots, \mathbf{a}_M]$.

The number of planets M is usually unknown. In that case, we divide the entire 2D space \mathcal{G} , defined as $\mathcal{G} = \{(e, f), 0 \leq e < e_{\max}, -\frac{f_{\max}}{2} < f < \frac{f_{\max}}{2}\}$, into a grid of prespecified size K . We point out here that the grid \mathcal{G} does not include the periastron parameter, which is taken to be zero for now but will be estimated as described later on. The choices of e_{\max} and f_{\max} in \mathcal{G} depend on the sampling pattern: to determine them, we calculate the spectral window defined as:

$$W(e, f) = \left| \frac{1}{N} \sum_{n=1}^N \exp(jv(t_n)) \right|^2, \quad 0 \leq e < 1, -\infty \leq f \leq \infty.\tag{10.7}$$

For any choice of (e, f) and t_n , there exists a $v(t_n)$ obtained via (10.2). Thus the spectral window W is a 2D function. Fig. 10.1 shows a typical sampling scheme for radial velocity measurements and its corresponding spectral window (in dB). The parameters e_{\max} and f_{\max} are chosen such that the region \mathcal{G} is unambiguous. By unambiguous, we mean that there are no repetitions in spectral window values inside \mathcal{G} . As can be seen from Fig. 10.1(b), in the region outside $\mathcal{G} = \{(e, f), 0 \leq e < 0.6, -\frac{1}{2} < f < \frac{1}{2}\}$, the spectral window contains approximate aliases and thus creates ambiguities. Consequently the parameters e_{\max} and f_{\max} are chosen in this example as 0.6 and 1 cycle/day, respectively.

The grid size is given by $K = K_1 \times K_2$, where $K_1 = \lfloor \frac{e_{\max}}{\Delta e} \rfloor$ and $K_2 = \lfloor \frac{f_{\max}}{\Delta f} \rfloor$; Δe and Δf denote the step sizes for eccentricity and frequency, respectively, and $\lfloor x \rfloor$ denotes the largest integer not exceeding x . A typical choice for Δe and Δf is $\Delta e = 0.01$ and $\Delta f = \frac{1}{5(t_N - t_1)}$. Following these preparations, we will use the data model in (10.6) with $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_K]$ and $\mathbf{s} = [\alpha_1, \dots, \alpha_K]^T$.

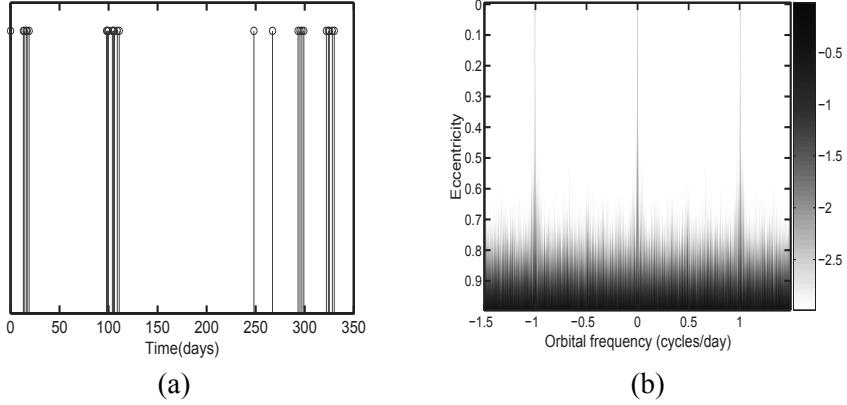


Figure 10.1. (a) Sampling pattern and (b) Spectral window (the different gray-levels represent the spectral window values on a decibel scale).

10.3 IAA based methods

10.3.1 IAA

Using the notation in (10.5), a weighted least squares fitting criterion, which lies at the basis of the IAA algorithm, can be defined as follows:

$$\min_{\alpha_m} \|\mathbf{y} - \mathbf{a}_m \alpha_m\|_{\mathbf{R}^{-1}}^2, \quad (10.8)$$

where $R = \sum_{m=1}^K |\alpha_m|^2 \mathbf{a}_m \mathbf{a}_m^H$. The solution to this problem is given by [98]:

$$\hat{\alpha}_m = \frac{\mathbf{a}_m^H R^{-1} \mathbf{y}}{\mathbf{a}_m^H R^{-1} \mathbf{a}_m}. \quad (10.9)$$

Since the matrix R depends on α_m , an iterative algorithm is used to estimate α_m and R . The steps of the IAA algorithm are summarized in Algorithm 1. The standard periodogram can be interpreted as the initializing step of the IAA algorithm (corresponding to $R = I$). IAA usually takes no more than 15 iterations to converge (see [122] [85]).

10.3.2 Regularized IAA methods

The matrix R in (10.9) can become ill-conditioned during the iterations or when the sampling pattern is pathological. In this section, we describe three different ways of regularization which differ only in the construction of the R matrix. At the i^{th} iteration, the R matrix in the different regularization methods is constructed as shown in the Table 10.1.

The noise covariance matrix is not always available. However in the case of radial velocity data, we typically have an estimate of the uncertainty in the

Algorithm 1 Iterative adaptive approach algorithm (IAA).

Initialization

Use the estimate in (10.9) with $R = I$ as the initial value.

Iteration

At the i^{th} iteration, the estimate of α_m is given by:

$$\hat{\alpha}_m^i = \frac{a_m^H(R^i)^{-1}y}{a_m^H(R^i)^{-1}a_m} \quad \text{for } m = 1, \dots, K$$

$$\text{where } R^i = \sum_{m=1}^K |\hat{\alpha}_m^{i-1}|^2 a_m a_m^H.$$

Termination

The iteration will be terminated when the relative change in $\hat{\alpha}_m$, $|\hat{\alpha}_m^i - \hat{\alpha}_m^{i-1}|^2$, is less than 10^{-4} .

The final spectral estimate at grid point m is given by $|\hat{\alpha}_m^c|^2$, where $\hat{\alpha}_m^c$ denotes the estimate obtained at convergence.

Regularization method	Covariance matrix (R)
IAA (with knowledge on noise covariances)	$R^i = \sum_{m=1}^K \hat{\alpha}_m^{i-1} ^2 a_m a_m^H + \sigma I$, where σ denotes the noise covariance matrix.
IAAR [85]	$R^i = \sum_{m=1}^K \hat{\alpha}_m^{i-1} ^2 a_m a_m^H + \sum_{l=1}^N \hat{\gamma}_l^{i-1} ^2 u_l u_l^H$, where u_l denotes the l^{th} column of the $N \times N$ identity matrix, and $\hat{\gamma}_l^i = \frac{u_l^H(R^i)^{-1}y}{u_l^H(R^i)^{-1}u_l}$.
VIAA	$R^i = \sum_{m \in \mathcal{M}_i} \hat{\alpha}_m^{i-1} ^2 a_m a_m^H + \sum_{m \in [1, K] \setminus \mathcal{M}_i} \hat{\alpha}_m^{i-1} ^2 I$, where \mathcal{M}_i denotes the subset of $[1, K]$ containing the indices corresponding to the N largest values of $\{ \hat{\alpha}_m^{i-1} ^2\}$ and I denotes the $N \times N$ identity matrix.

Table 10.1. Regularized IAA methods

data that can be used for regularization by means of a diagonal σ matrix with the noise variances on the diagonal. In the case of IAAR, at convergence the values $\{|\hat{\gamma}|^2\}_{l=1}^N$ will be close to the noise variances. The regularization in VIAA is based on the intuitive fact that from N data samples we can reliably estimate no more than N values in the spectrum. Thus the N largest values in the spectrum are used to construct the signal part of R and the sum of the remaining values in the spectrum is taken as an estimate of the noise floor.

Next, we consider a simulated data example to illustrate the performance of the IAA algorithm. The data set consists of 120 nonuniformly placed samples spanning 1600 days. The sampling pattern and the data are shown in Fig. 10.2. From the spectral window, the parameters e_{\max} and f_{\max} are calculated to be 0.6 and 1 cycle/day respectively. The grid size K is chosen to be 6×10^4 , i.e. $K_1 = 60$ and $K_2 = 10^3$. The data set is made up of two planets with $(e_1 = 0.2, f_1 = 0.23 \text{ cycles/day}, T_1 = 4 \text{ days})$ and $(e_2 = 0, f_2 = 0.2505 \text{ cycles/day}, T_2 = 3 \text{ days})$, and radial velocity amplitudes 6 and 4 respectively. The planet at frequency 0.2505 cycles/day is chosen deliberately such that it does not lie on the grid. We also note here that the planet configuration mentioned above is not necessarily representative of typical, stable planetary systems, as their orbital period separation is less than 10%. Nevertheless we have chosen this example to illustrate the performance of the proposed methodology on a challenging case with a small frequency separation. The noise in the data is taken to be white Gaussian with zero-mean and variance equal to 0.5. Fig. 10.3 shows the spectral estimates obtained from the periodogram and from IAA. It is evidenced by the figure that while the periodogram suffers from heavy leakage and fails to accurately locate the planets present in the data, IAA estimates well at least one of the planets. The reason for IAA failing to locate the planet at 0.2505 cycles/day is the choice of a coarse grid to speed up the computation. In the next section, we will show how to refine the spectral estimates when the grid is coarse.

10.3.3 Computational aspects and Range-selective IAA

In each iteration of the IAA algorithm, the construction of the R matrix, formed as a sum of K rank-one matrices, requires $O(KN^2)$ flops (one flop = one complex multiplication and one complex addition). Then the inversion of R , done once for each iteration, requires $O(N^3)$ flops. A major computational requirement in each iteration comes from the calculation of $\hat{\alpha}_m$, which requires a matrix-vector multiplication $R^{-1}a_m$. Each of these matrix-vector multiplications requires $O(N^2)$ flops and thus the calculation of all estimates $\{\hat{\alpha}_m\}_{m=1}^K$ requires $O(KN^2)$ flops. Therefore in each iteration, IAA requires $O(KN^2 + N^3)$ flops. Since the number of iterations is usually quite small (the performance of IAA does not change significantly after 15 iterations), it follows that $O(KN^2 + N^3)$ flops is the total complexity of the IAA algorithm.

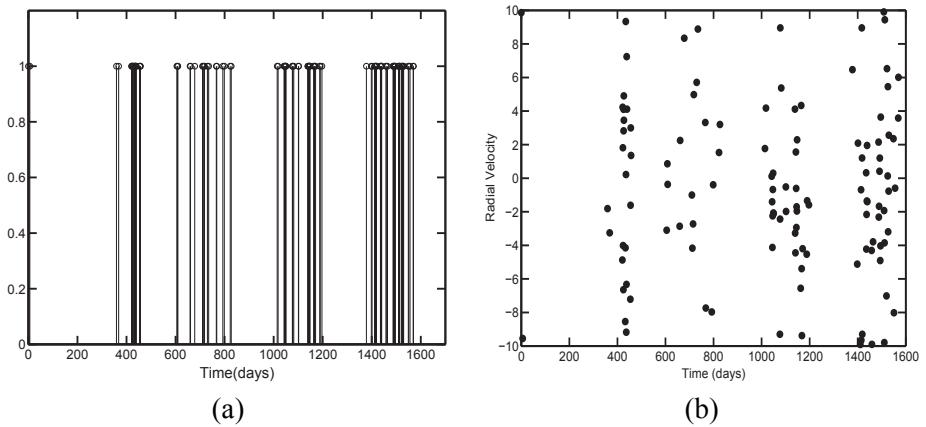


Figure 10.2. (a) Sampling pattern and (b) data.

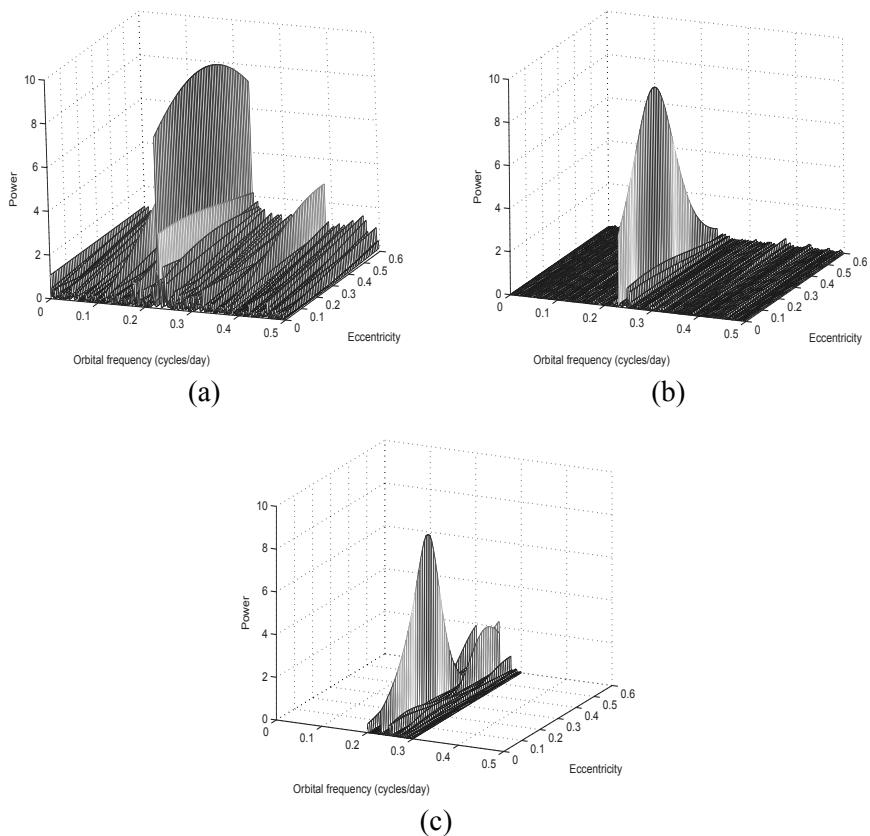


Figure 10.3. Spectral estimates obtained from (a) Periodogram (b) IAA (c) RSIAA.

From the analysis above, we see that the IAA method can be computationally expensive when the grid size is large. However, if we are interested in only certain ranges of frequency and eccentricity, the computations can be reduced drastically by considering the following range-selective IAA (RSIAA) algorithm. In RSIAA, only the powers of the components present in the region of interest (ROI) are calculated. To suppress the interferences, the N largest values of the periodogram from the region outside ROI are included in the construction of R . Let $\mathcal{L} \subset \mathcal{G}$ denote the grid points in ROI, and let $\mathcal{T} \in \{\mathcal{G} \setminus \mathcal{L}\}$ denote the indices of the N largest values of the periodogram in the region outside ROI. Algorithm 2 summarizes the RSIAA algorithm.

For the simulated example considered above, we focus on the frequency range of 0.2 cycles/day to 0.3 cycles/day and use RSIAA to obtain the spectrum within the ROI. As can be seen from Fig. 10.3(c), the RSIAA algorithm also accurately locates the component at 0.23 cycles/day but with less computations (it is roughly 10 times faster than IAA). As to the eccentricity, the maximum power at frequency 0.23 cycles/day occurs at 0.22, which is close to the true value. Therefore, RSIAA successfully detects the planet with orbital frequency of 0.23 cycles/day. Since the other planet with orbital frequency 0.2505 is off the grid, we need further steps that are discussed below to detect it.

Algorithm 2 Range-selective IAA (RSIAA).

Initialization

Use the estimate in (10.9) with $R = I$ as the initial value.

Iteration

At the i^{th} iteration, the estimate of α_m is given by:

$$\hat{\alpha}_m^i = \frac{a_m^H(R^i)^{-1}y}{a_m^H(R^i)^{-1}a_m} \quad \text{for } m \in \{\mathcal{L} \cup \mathcal{T}\}$$

$$\text{where } R^i = \sum_{m \in \mathcal{L} \cup \mathcal{T}} |\hat{\alpha}_m^{i-1}|^2 a_m a_m^H.$$

Termination

The iteration will be terminated when the relative change in $\hat{\alpha}_m$, $|\hat{\alpha}_m^i - \hat{\alpha}_m^{i-1}|^2$, is less than 10^{-4} .

10.4 Refining the estimates and statistical significance testing: RELAX and GLRT

Let $S(e_l, f_m)$ denote the IAA power estimate at the grid point k , i.e. $S(e_l, f_m) = |\hat{\alpha}_k|^2$, where $l = \left\lfloor \frac{k-1}{K_2} \right\rfloor + 1$ and $m = k - (l-1)K_2$. Then for any m , $1 \leq m \leq K_2$, we calculate

$$\begin{aligned} \tilde{l} &= \arg \max_{1 \leq l \leq K_1} S(e_l, f_m). \\ \tilde{S}_m &= S(e_{\tilde{l}}, f_m) \end{aligned} \tag{10.10}$$

Therefore \tilde{S}_m ($\tilde{\alpha}_m$) denotes the maximum power (amplitude) at frequency f_m over all possible eccentricities. We then pick the $5 - 10$ largest peaks in $\{\tilde{S}_m\}_{m_1}^{m_2}$, where $m_1 = \left\lfloor \frac{K_2}{2} \right\rfloor + 1$ and $m_2 = K_2$. The choice of m_1 and m_2 indicates that the peaks are chosen from the positive part of the IAA spectrum; also in the case of radial velocity data, the choice of $5 - 10$ peaks appears to be reasonable for most applications. The values of periastron passage times corresponding to the selected peaks are estimated by running IAA (with VIAA regularization) on those peaks with an additional unknown parameter T . For a peak with frequency f and eccentricity e , a uniform grid (named as T -grid) of size 100 with values of T in the range $\left[0, \frac{1}{f}\right]$ is chosen and the IAA estimates are computed. Finally, the value of T for a given peak is obtained by choosing the T that corresponds to the maximum power over the T -grid associated with that peak.

The estimates obtained from IAA are usually fairly accurate. However, if the grid (\mathcal{G}) is not chosen fine enough (to reduce the computation time), then IAA might miss some true peaks. In that case, applying RELAX [55], a parametric iterative estimation algorithm, can refine the IAA estimates. Algorithm 3 briefly describes the steps involved in RELAX. The peaks picked from IAA are used as initial estimates for RELAX, which has a beneficial effect on the convergence of RELAX compared with using other more arbitrary initial estimates. RELAX generally converges within a few iterations (typically in less than 10 iterations).

Next we note that, under the assumption made on the noise in (10.4), the RELAX estimates are optimal in the maximum likelihood sense [55]. We can then use the generalized likelihood ratio test (GLRT) to establish the statistical significance of the estimated planet parameters. We first apply RELAX to the largest IAA peak and use GLRT to test the null hypothesis that there are no planets (or, in other words, that the data set is made only of white noise) against the hypothesis that there is at least one exoplanet. If the test rejects the null hypothesis then we will proceed and apply RELAX to the two largest peaks and subsequently test the hypothesis that there is one exoplanet in the data against the hypothesis that there are at least two exoplanets; and so on. As an example, for the following hypotheses

H_0 : There are no planets.

H_1 : There is at least one exoplanet with eccentricity \hat{e}_1 , orbital frequency \hat{f}_1 and periastron passage time \hat{T}_1 .

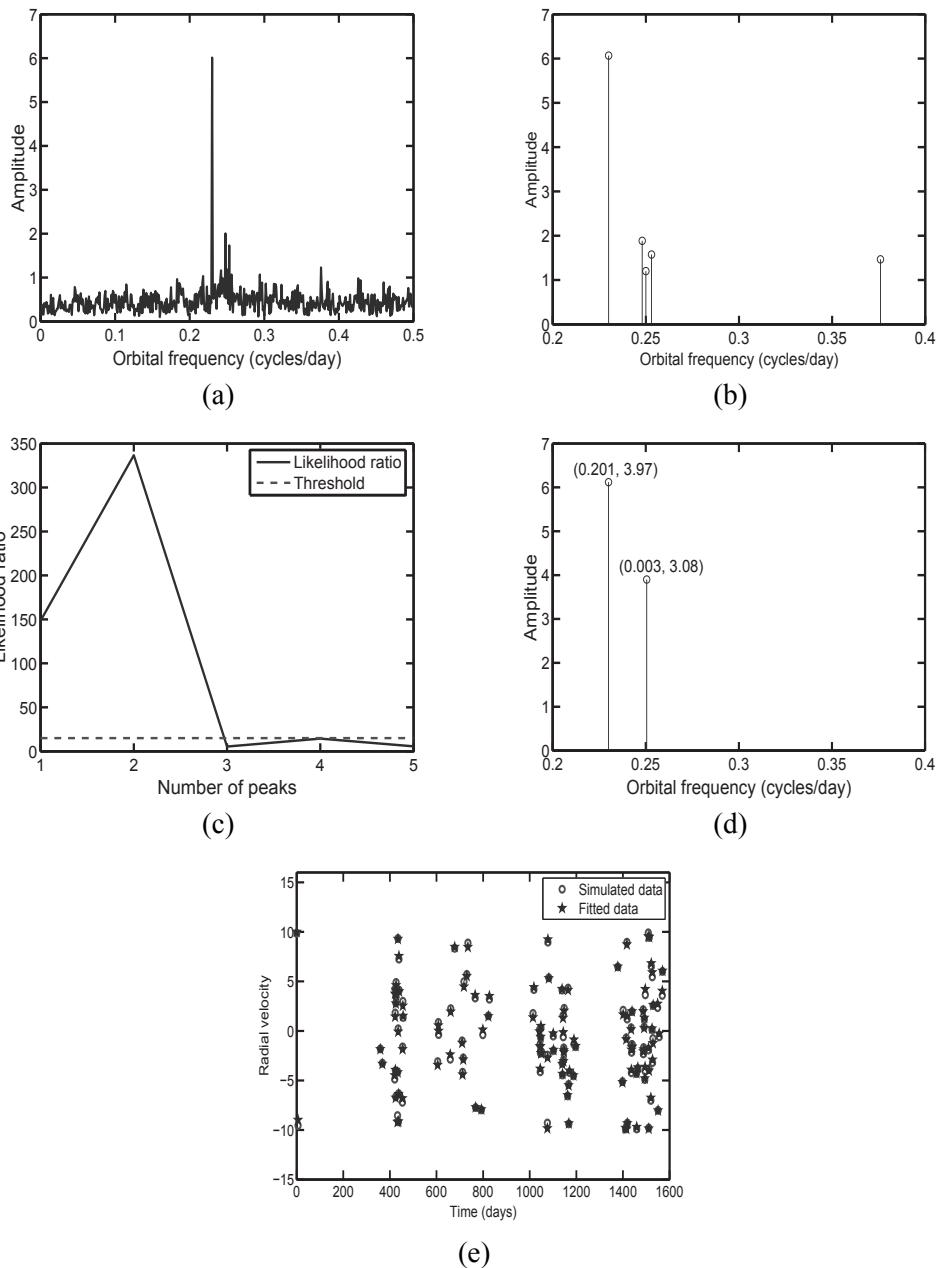


Figure 10.4. (a) the IAA spectrum. (b) the 5 largest peaks in the IAA spectrum. (c) likelihood ratio vs. the peak number. (d) RELAX estimates of the peaks. (e) comparison of the simulated and fitted data sequences. In (d), the eccentricity and the periastron passage time (in days) associated with a peak are indicated at the top of the corresponding peak.

the log-likelihood (LL) functions are given by:

$$\begin{aligned} \text{LL}(H_0) &= -\frac{N}{2} \ln \left(\sum_{n=1}^N |y(t_n)|^2 \right) + C, \\ \text{LL}(H_1) &= -\frac{N}{2} \ln \left(\sum_{n=1}^N |y(t_n) - \hat{r}_1 \cos(v(t_n)) - \hat{q}_1 \sin(v(t_n))|^2 \right) + C \end{aligned} \quad (10.11)$$

where C is an additive constant, $v(t_n)$ is calculated from the RELAX estimates $(\hat{e}_1, \hat{f}_1, \hat{T}_1)$, and \hat{r}_1, \hat{q}_1 are the least square estimates of r, q corresponding to $(\hat{e}_1, \hat{f}_1, \hat{T}_1)$, see Algorithm 3. Under the assumption that hypothesis H_0 is true, the log-likelihood-ratio, defined as $2(\text{LL}(H_1) - \text{LL}(H_0))$, is asymptotically a random variable with a chi-square distribution. Then the GLRT is given by

$$2(\text{LL}(H_1) - \text{LL}(H_0)) \stackrel[H_1]{>=}{H_0} \Lambda \quad (10.12)$$

where Λ denotes a fixed threshold. The threshold is usually chosen such that $\text{prob}(X \leq \Lambda) = \xi$, where $X \sim \chi_5^2$ denotes a chi-square distributed random variable with 5 degrees of freedom (because of the 5 unknowns per planet in the data model, namely e, f, T, r and q), and ξ determines the significance level of the test. Choosing $\xi = 0.99$ gives a false alarm probability of 0.01 and the corresponding threshold is $\Lambda = 15$.

For the example discussed in the last section, Fig.10.4(a) shows the IAA spectrum ($\tilde{\alpha}_m$ vs. frequency) and Fig.10.4(b) shows the 5 largest peaks in the IAA spectrum. Fig.10.4(c) shows the plot of likelihood ratio versus the number of peaks. It can be concluded from Fig.10.4(c) that there are two planets present in the data, and Fig.10.4(d) shows the RELAX estimates of frequencies, eccentricities and periastron passage times for those two planets. As seen from Fig.10.4(b) and 10.4(d), IAA combined with RELAX locates both components at 0.23 cycles/day and 0.2505 cycles/day. Thus the combination of IAA with RELAX has high resolution and estimates accurately both the frequencies as well as the eccentricities and the periastron passage times of the components in the data.

10.5 Real life radial velocity data sets

In this section, we consider the application of the algorithm introduced in the previous sections to real life radial velocity data. Our goal is to detect the exoplanets present in a star system and estimate their eccentricities, frequencies and periastron passage times. We will consider radial velocity data sets for three different stars:

HD 63454: a class K type star with a planet in a nearly circular orbit according to [72].

HD 208487: a class G type star with a planet in an elliptical orbit according to

Algorithm 3 RELAX

Let $\{(e_p^0, f_p^0, T_p^0)\}_{p=1}^P$ denote the parameters of the P most dominant planets in the IAA spectrum (e.g. $P = 5 - 10$). The initial values of their radial velocity amplitudes $\{(r_p^0, q_p^0)\}_{p=1}^P$ are taken to be zero.

for $p = 1$ to P **do**
 for $i = 1$ to I (e.g. $I = 10$) **do**
 for $u = 1$ to p **do**
 $y_u^i(t_n) = y(t_n) - \sum_{k=1, k \neq u}^p (r_k^{i-1} \cos(v_k^{i-1}(t_n)) + q_k^{i-1} \sin(v_k^{i-1}(t_n)))$
 where v_k^{i-1} is obtained from $(e_k^{i-1}, f_k^{i-1}, T_k^{i-1})$. Then
 $(e_u^i, f_u^i, T_u^i) = \arg \min_{\{e, f, T\}} \min_{\{r, q\}} \sum_{n=1}^N |y_u^i(t_n) - r \cos(v(t_n)) - q \sin(v(t_n))|^2$
 The inner minimization with respect to $\{r, q\}$ in the above equation is a least squares problem and the estimates $\{r_u^i, q_u^i\}$ can be obtained analytically (see [98]). The minimization with respect to $\{e, f, T\}$ is carried out via a 3D grid search performed around $(e_u^{i-1}, f_u^{i-1}, T_u^{i-1})$.
 end for
 end for
 for $u = 1$ to p **do**
 $(e_u^0, f_u^0, T_u^0) \leftarrow (e_u^I, f_u^I, T_u^I)$ and $(r_u^0, q_u^0) \leftarrow (r_u^I, q_u^I)$.
 end for
 end for
 $\{(\hat{e}_p, \hat{f}_p, \hat{T}_p) \leftarrow (e_p^I, f_p^I, T_p^I)\}_{p=1}^P$ and $\{(\hat{r}_p, \hat{q}_p) \leftarrow (r_p^I, q_p^I)\}_{p=1}^P$

Planet No.	Previous work				This work			
	e	$f(\text{cycles/day})$	$T(\text{days})$	β	e	$f(\text{cycles/day})$	$T(\text{days})$	β
1	0	0.3549	0.2778	0.0643	0.0130	0.3549	0.3595	0.0634

Table 10.2. Parameters of the planet of HD 63454 star system.

[108].

GJ 876: a class M type star with two planets in a circular orbit according to [64], [81] and one planet in an elliptical orbit, see [63].

These examples are considered to be representative of typical real life cases. In each example, we will show the following plots:

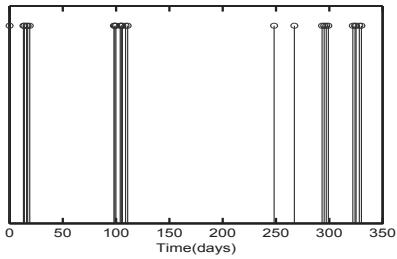
- Sampling pattern and the radial velocity data.
- Amplitude vs. orbital frequency for IAA and RELAX (eccentricity and periastron passage time values for the peaks in the amplitude spectrum are indicated in the plots).
- Likelihood ratio vs. the planet number.
- Observed and fitted data sequences.

10.5.1 HD 63454

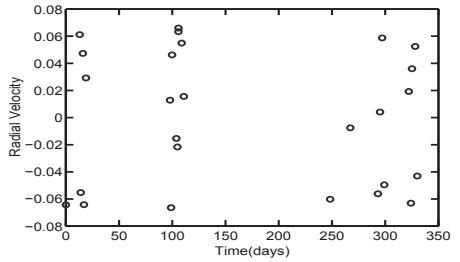
The radial velocity data set used in this example contains 26 samples spanning 350 days. The sampling pattern and the data are as shown in Figs.10.5(a) and 10.5(b). The parameters e_{\max} and f_{\max} are determined from the spectral window to be 0.6 and 1. The IAA algorithm is then applied to the data and the largest 5 peaks in the IAA spectrum are picked, as shown in Fig.10.5d. VIAA is used here as the regularization method. The eccentricity and the periastron passage time (in days) associated with a peak are indicated at the top of the corresponding peak. Finally, RELAX is applied to the peaks picked from the IAA spectrum. Fig.10.5(e) shows the likelihood ratio vs. the peak number, and it can be concluded that there exists only one planet in the HD 63454 star system. Fig. 10.5(f) shows the RELAX estimates of the frequency, eccentricity and the periastron passage time of the detected planet. The estimated frequency of 0.3549 cycles/day (orbital period of 2.8176 days), the eccentricity of 0.0130 (nearly circular orbit) and the periastron passage time of 0.3595 days match closely the reported values in [72], see Table 10.2.

10.5.2 HD 208487

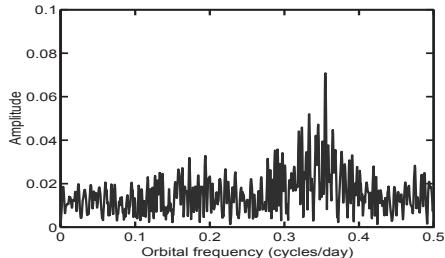
The data set used next consists of 31 samples spanning 2250 days, as is shown in Fig.10.6(b). The parameters e_{\max} and f_{\max} are determined from the spectral window to be 0.5 and 1 cycles/day. The spectrum obtained using IAA is shown in Fig.10.6(c), which indicates the presence of more than one planet.



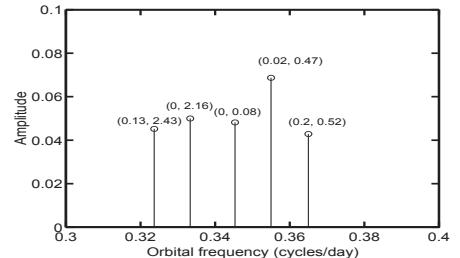
(a)



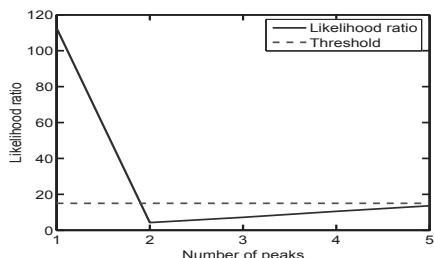
(b)



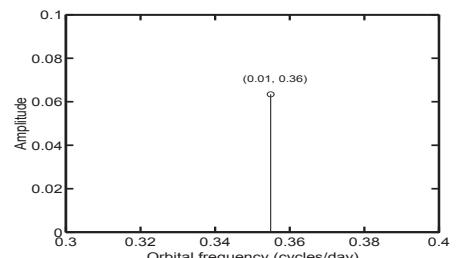
(c)



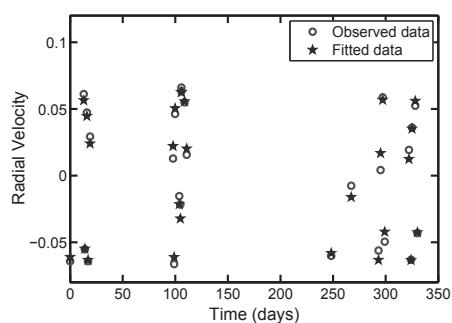
(d)



(e)



(f)



(g)

Figure 10.5. HD 63454: (a) sampling pattern, (b) radial velocity data, (c) the IAA spectrum, (d) the 5 largest peaks in the IAA spectrum, (e) the likelihood ratio, (f) the RELAX spectrum and (g) comparison of the observed and fitted data sequences.

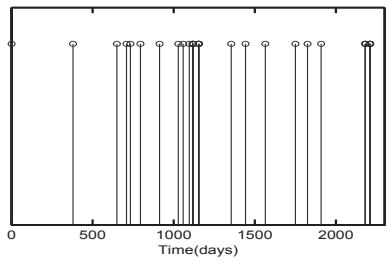
Planet No.	Previous work				This work			
	e	$f(\text{cycles/day})$	$T(\text{days})$	β	e	$f(\text{cycles/day})$	$T(\text{days})$	β
1	0.24	0.0077	92	20	0.3260	0.0078	130.9	19.9
2	-	-	-	-	0.3150	0.0690	14.2	12.18
3*)	-	-	-	-	0	0.0408	2.9	4.96

Table 10.3. Parameters of the planets of HD 208487 star system. *) The third planet becomes statistically insignificant if the false alarm probability is decreased from 10^{-2} to 10^{-4} , in which case the threshold becomes $\Lambda = 25$.

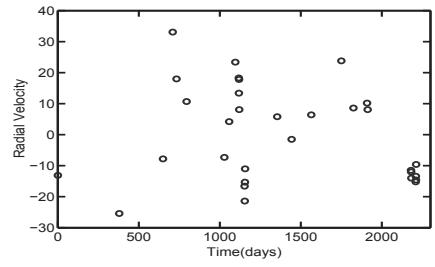
The 5 largest peaks in the IAA spectrum are picked up and are used to initialize RELAX. The GLRT plot shown in Fig.10.6(e) suggests the existence of three planets in the HD 208487 star system with the following parameters (see Fig.10.6(f) and also Table 10.3): ($e_1 = 0.326$, $f_1 = 0.0078$ cycles/day, $T_1 = 130.9$ days), ($e_2 = 0.315$, $f_2 = 0.069$ cycles/day, $T_2 = 14.2$ days) and ($e_3 = 0$, $f_3 = 0.0408$ cycles/day, $T_3 = 2.9$ days). However [108] reported that the star has only one planet with an orbital frequency of 0.0078 cycles/day. Fig. 10.6(g) and 10.6(h) show the plots of measured data and the fitted data obtained assuming the existence of one and, respectively, three planets. It is seen clearly from these figures that the three planet model fits the measured data much better than a single planet model. However, besides the evidence provided by the GLRT statistical testing and by the significantly better fitting of the observed data, we do not have other evidence to support the claim on two additional planets for this star system.

10.5.3 GJ 876

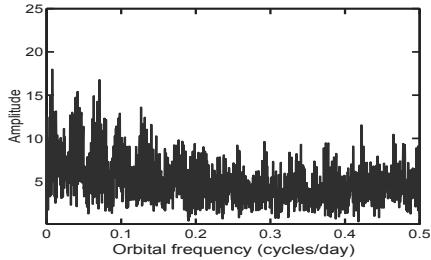
The data set for star GJ 876 consists of 100 samples spanning 2000 days. The sampling pattern and the radial velocity data are shown in Fig.10.7(a) and Fig.10.7(b). The parameters e_{\max} and f_{\max} are determined to be, at least approximately, 0.5 and 1 cycle/day. The spectrum obtained using IAA is shown in Fig.10.7(c). From the IAA spectrum, the 10 largest peaks are picked up (see Fig.10.7(d)) and they are used as initial estimates for RELAX. The GLRT indicates the existence of five planets, see Fig.10.7(e), whose estimated parameters are shown in Fig.10.7(f) and Table 10.4. However, the component with an orbital frequency of 0.0168 cycles/day is probably unphysical, as it is likely caused by the precession of the orbit of the planet with the orbital frequency of 0.0164 cycles/day, due to planet-planet interactions that make the actual motion deviate from the Keplerian orbit. Note that a previous study by [81] reported the presence of only three planets with orbital frequencies 0.0164 cycles/day, 0.033 cycles/day and 0.516 cycles/day. However, this three planet model is questionable from a statistical viewpoint, as it corresponds to a local (rather than a global) maximum of the likelihood function; indeed, we have found that there are other three planet models that



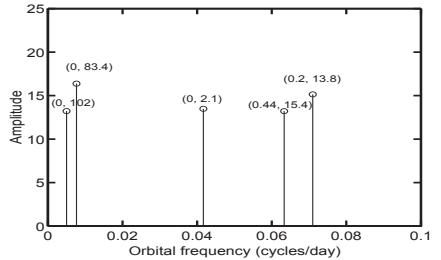
(a)



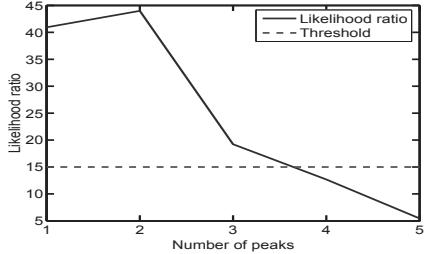
(b)



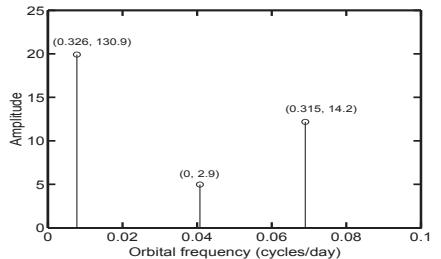
(c)



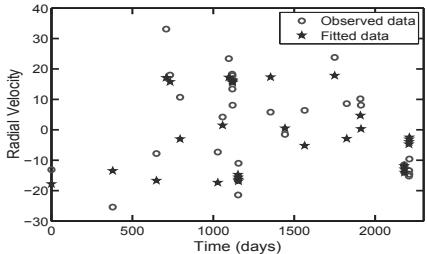
(d)



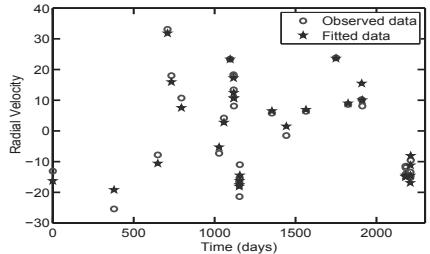
(e)



(f)



(g)



(h)

Figure 10.6. HD 208487: (a) sampling pattern, (b) radial velocity data, (c) the IAA spectrum, (d) the 5 largest peaks in the IAA spectrum, (e) the likelihood ratio, (f) the RELAX spectrum and (g) comparison of the observed data and fitted data obtained from the parameters of the planet reported in [108]. (h) comparison of the observed data and fitted data obtained from the three detected planets.

Planet No.	Previous work				This work			
	e	f (cycles/day)	T (days)	β	e	f (cycles/day)	T (days)	β
1	0.036	0.0164	0	212.60	0	0.0164	0.0017	215.09
2	0.224	0.0331	2.7	88.36	0	0.0331	0	82.62
3*)	0	0.516	0.73	6.4	0	0.0011	0	9.61
4*)	-	-	-	-	0.04	0.0066	27.01	9.95
**)	-	-	-	-	0.245	0.0168	0.029	23.57

Table 10.4. Parameters of the planets of GJ 876 star system. *) The fourth planet becomes statistically insignificant and the third planet becomes nearly statistically insignificant if the false alarm probability is decreased from 10^{-2} to 10^{-4} , in which case the threshold becomes $\Lambda = 25$. **) Probably unphysical component due to planet-planet interactions.

give larger values of the likelihood than the model of [81]. For example, the three planet model with parameters ($e_1 = 0$, $f_1 = 0.0164$ cycles/day, $T_1 = 0$ days, $\beta_1 = 210.75$), ($e_2 = 0.047$, $f_2 = 0.0331$ cycles/day, $T_2 = 0.3283$ days, $\beta_2 = 79.58$) and ($e_3 = 0$, $f_3 = 0.0023$ cycles/day, $T_3 = 0.3391$ days, $\beta_3 = 14.9$) has a log-likelihood value of -510.2667, which is higher than the value of -517.4751 for the model of [81]. Concerning a comparison of the three planet model of [81] and the planet model proposed here, Fig.10.7(g) and 10.7(h) show that the latter fits the observed data significantly better than the former. However, the same cautionary remark, that we made at the end of the previous sub-section, applies here as well.

10.6 Conclusions

The simulated and real life examples discussed in the chapter suggest that our algorithm successfully detects the presence of spectral peaks (planets) in radial velocity data and accurately identifies both their frequencies and eccentricities as well as their periastron passage times. The examples used here are typical of cases usually encountered in exoplanet search and hence the proposed algorithm is believed to be an effective and useful tool.

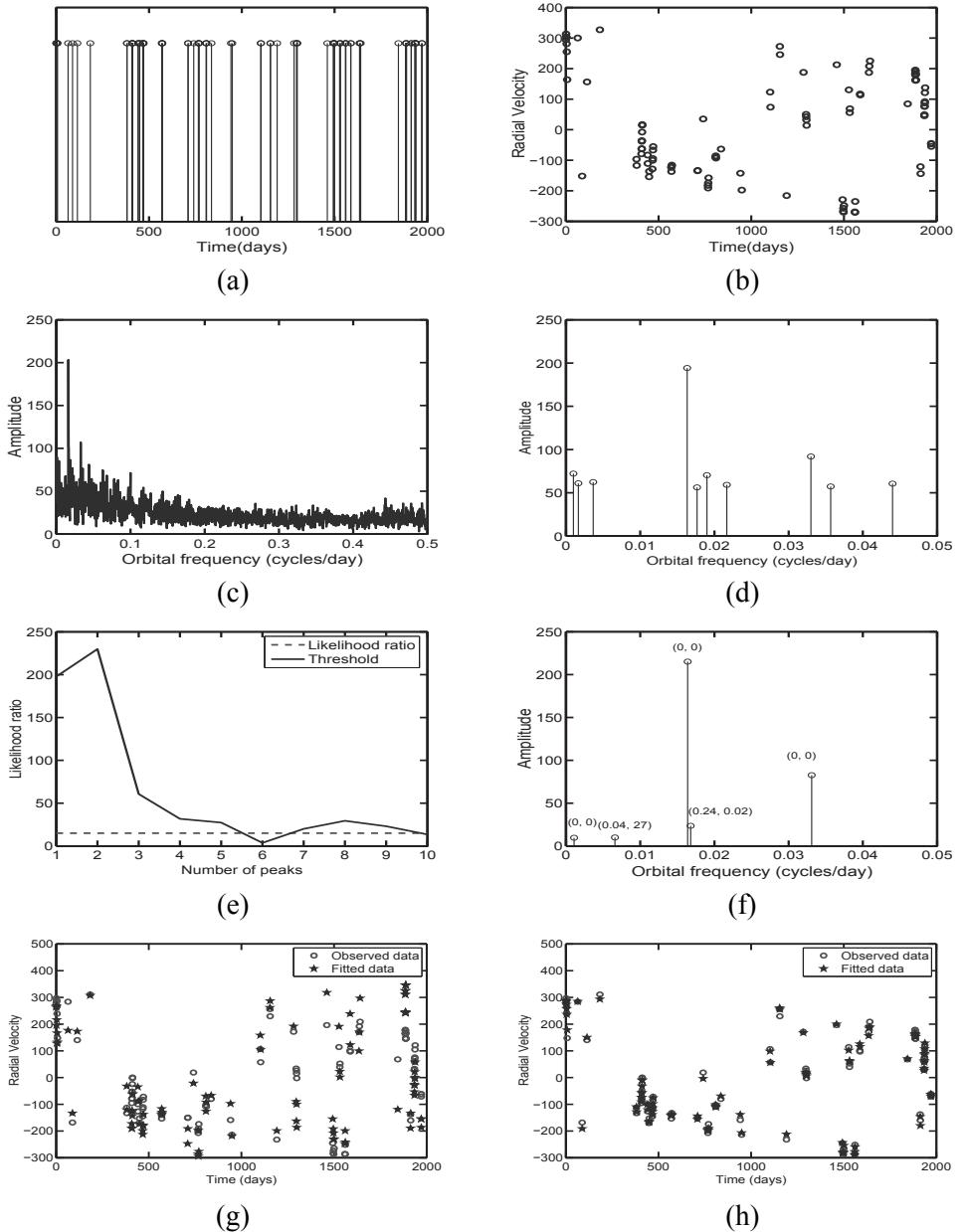


Figure 10.7. GJ 876: (a) sampling pattern, (b) radial velocity data, (c) the IAA spectrum, (d) the 10 largest peaks in the IAA spectrum, (e) the likelihood ratio, (f) the RELAX spectrum, (g) comparison of the observed data and the fitted data obtained from the three planets reported in [81] and (h) comparison of the observed data and fitted data obtained from the 4 detected planets and the 5th (unphysical) component.

11. A combined linear programming-maximum likelihood approach to radial velocity data analysis

11.1 Introduction and the data model

Extrasolar planet (or, shortly, exoplanet) detection is a fascinating and challenging field of research in the area of astronomy and astrophysics. Out of the many techniques employed in exoplanet search, the radial velocity (RV) technique has been the most widely used one. In this technique the wobbling of the parent star, due to the possible presence of planet(s), is captured by measuring the Doppler shifts of the light from the star. The measured shifts, and hence the radial velocities of the star, are analyzed to determine the presence of planet(s) and to estimate their orbital parameters.

The Lomb-Scargle or Least-Squares periodogram (LSP) [86] has been the most commonly used method for analysis of RV data. However, LSP suffers from leakage problems which are inherent to periodogram based methods, see, e.g. [98]. Moreover, LSP can handle only planets with circular orbits. In the case of planets with elliptical orbits, the orbital parameters are generally estimated by direct minimization of a highly nonlinear least squares (NLS) criterion obtained by fitting the Keplerian model to the RV data, see below for details.

In the following we briefly introduce the Keplerian model for RV data. Let $\{y(t_n)\}_{n=1}^N$ denote the radial velocity of a star measured at a set of possibly nonuniform time instants $\{t_n\}_{n=1}^N$. According to the Keplerian model of planetary motion, see e.g. [5], the radial velocity data can be modeled as follows:

$$y(t_n) = K_0 + \sum_{m=1}^M K_m [\cos(\omega_m + v_m(t_n)) + e_m \cos(\omega_m)], \quad (11.1)$$

$$n = 1, \dots, N$$

where

$$\tan(v_m(t_n)) = \sqrt{\frac{1+e_m}{1-e_m}} \tan(E_m(t_n)) \quad (11.2)$$

$$E_m(t_n) - e_m \sin(E_m(t_n)) = \frac{2\pi(t_n - T_m)}{P_m},$$

and

- K_0 : Constant radial velocity.
- M : Number of exoplanets revolving around the star.
- e_m : Eccentricity of the orbit of the m^{th} planet.
- ω_m : Longitude of the periastron for the m^{th} planet.

- P_m : Orbital period of the m^{th} planet, related to orbital frequency f_m by $f_m = \frac{1}{P_m}$.
- T_m : Periastron passage time of the m^{th} planet.
- K_m : Radial velocity amplitude of the m^{th} planet.
- $v_m(t_n), E_m(t_n)$: True and eccentric anomaly of the m^{th} planet, with t_n indicating their time dependence.

A simple calculation shows that (11.1) can be re-written as :

$$y(t_n) = \sum_{m=0}^M \alpha_m \cos(v_m(t_n)) + \sum_{m=1}^M \beta_m \sin(v_m(t_n)) \quad (11.3)$$

where $\alpha_0 = K_0 + \sum_{m=1}^M K_m e_m \cos(\omega_m)$, $\alpha_m \stackrel{\Delta}{=} K_m \cos(\omega_m)$, $\beta_m \stackrel{\Delta}{=} -K_m \sin(\omega_m)$, and $v_0(t_n) = 0$. Note that from α_m and β_m in (11.3) we can easily determine the orbital parameters K_m and ω_m as :

$$K_m = \sqrt{\alpha_m^2 + \beta_m^2} \quad \omega_m = \arctan\left(\frac{-\beta_m}{\alpha_m}\right) \quad (11.4)$$

The model (11.3) can be expressed more compactly making use of the following notation:

$$\begin{aligned} \mathbf{y} &= [y(t_1), \dots, y(t_N)]^T, \\ \mathbf{a}_m &= [\cos(v_m(t_1)), \dots, \cos(v_m(t_N))]^T, \\ \mathbf{b}_m &= [\sin(v_m(t_1)), \dots, \sin(v_m(t_N))]^T. \end{aligned} \quad (11.5)$$

Using this notation and simultaneously accounting for the noise in the measurements, (11.3) can be re-written as follows:

$$\mathbf{y} = \mathbf{C}^T \mathbf{s} + \mathbf{e} \quad (11.6)$$

where

$$\begin{aligned} \mathbf{C}^T &\stackrel{\Delta}{=} [\mathbf{c}_1 \quad \dots \quad \mathbf{c}_{2M+1}] \\ &= [\mathbf{a}_0 \quad \dots \quad \mathbf{a}_M \quad \mathbf{b}_1 \quad \dots \quad \mathbf{b}_M], \end{aligned}$$

$\mathbf{s} \stackrel{\Delta}{=} [s_1, \dots, s_{2M+1}]^T = [\alpha_0, \dots, \alpha_M, \beta_1, \dots, \beta_M]^T$, and $\mathbf{e} = [e(t_1), \dots, e(t_N)]^T$. The sequence $\{e(t_n)\}_{n=1}^N$ is assumed to be a zero-mean white noise sequence with variances $\sigma_1, \dots, \sigma_N$.

The number of planets M and their corresponding orbital parameters (e, f, T) are unknown and have to be estimated from the measurements $\{y(t_n)\}$. With this fact in mind, we divide the 2D set of possible values for e and f , defined as $\mathcal{G} = \{(e, f), 0 \leq e < e_{\max}, 0 < f < f_{\max}\}$, using a grid of prespecified size Q . The parameter T is not included in this grid and will be estimated as explained later on. The choice of e_{\max} and f_{\max} depends on the sampling pattern and can be based on the spectral window, see, e.g. [5] for more details.

In this chapter we apply a new technique named SPICE to the analysis of RV data. SPICE enjoys global convergence and yet it does not require the selection of any user parameters (aka hyperparameters). In effect SPICE can

be implemented as a linear program (LP) that can be solved efficiently. The orbital parameters estimated by SPICE are then refined by running the RELAX algorithm [5] [55] using the SPICE estimates as initial values, and finally the significance of the resultant estimates is established by carrying out the GLRT.

In Section 11.2 we introduce the SPICE technique and discuss its different formulations. Section 11.3 discusses the RELAX algorithm and the GLRT. In Section 11.4 we apply our technique to a real-life data set and compare the results with those reported in the literature. Section 11.5 then draws the conclusions of the chapter.

11.2 SPICE

The SPICE technique obtains estimates of $\{s_k\}$ and $\{\sigma_k\}$ by solving the following optimization problem (we refer the reader to [93] for a statistical motivation of the constrained fitting criterion below) :

$$\min_{\{\tilde{p}_k \geq 0\}} y^T R^{-1} y \quad \text{s.t. } \sum_{k=1}^{2Q+N+1} \frac{\|c_k\|^2}{\|y\|^2} \tilde{p}_k = 1 \quad (11.7)$$

where

$$\begin{aligned} R &= E(yy^T) = \sum_{k=1}^{2Q+1} s_k^2 c_k c_k^T + \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_N) \\ &\stackrel{\Delta}{=} \tilde{C}^T \tilde{P} \tilde{C} \end{aligned} \quad (11.8)$$

and where

$$\begin{aligned} \tilde{C}^T &= [c_1, \dots, c_{2Q+1} I] \stackrel{\Delta}{=} [c_1, \dots, c_{2Q+N+1}] \\ \tilde{P} &= \text{diag}(s_1^2, s_2^2, \dots, s_{2Q+1}^2, \sigma_1, \dots, \sigma_N) \\ &\stackrel{\Delta}{=} \text{diag}(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_{2Q+1}, \tilde{p}_{2Q+2}, \dots, \tilde{p}_{2Q+N+1}) \end{aligned} \quad (11.9)$$

Hereafter $\|c_k\|$ and $\|y\|$ denote the Euclidean norms of c_k and y , respectively, I denotes the $N \times N$ identity matrix, and $\text{diag}(x_1, \dots, x_N)$ denotes the diagonal matrix with diagonal entries x_1, \dots, x_N . The second equality in (11.8) follows from the assumption that c_k is uncorrelated to c_l for any $k \neq l$. While this assumption may not hold exactly, especially when the grid points are closely spaced, the performance loss due to making it can be shown to be small, see, e.g. [98] [93]. Once $\{\tilde{p}_l\}$ are obtained, the actual power estimates denoted by $\{p_k\}$ are obtained as follows:

$$\begin{aligned} p_1 &= \tilde{p}_1 \\ p_k &= \tilde{p}_k + \tilde{p}_{k+Q} \quad k = 2, \dots, Q+1. \end{aligned}$$

The parameters of interest (e, f, T) (note that $K_l^2 = p_l$) are then obtained from $\{p_k\}$ as explained in the next section. The minimization problem in (11.7) is convex and can be re-written as a semidefinite program (SDP), see [93], which can be solved globally. However, for a grid size of large dimension, say $Q = 10^5$ (which is quite often the case), solving the said SDP could be

quite time consuming. To tackle this issue, in [93] we derived an iterative multiplicative algorithm (MA) that solves the minimization problem in (11.7) much faster than via SDP. More interestingly, by Elfving theorem, a standard result in optimal experiment design literature, see [89] for a simple proof, the optimization problem in (11.7) can be re-formulated as the following linear program :

$$\min_{\tilde{s}} \|\tilde{s}\|_1 = \sum_{k=1}^{2Q+N+1} |\tilde{s}_k| \quad \text{s.t. } \tilde{C}^T \tilde{s} \triangleq [C^T, I] \begin{bmatrix} s \\ e \end{bmatrix} = y \quad (11.10)$$

where $\|\tilde{s}\|_1$ denotes the ℓ_1 norm of \tilde{s} .

Linear programs, in general, can be solved much faster than SDP. Furthermore, the LP formulation of SPICE is faster (≈ 3 times for the case discussed in the numerical section) than the MA formulation of SPICE. In fact, the LP formulation of SPICE does not require any initialization, whereas the MA requires a reasonable initialization for fast convergence. Thus, to summarize, the SPICE optimization problem in (11.7) can be solved by three different means, namely SDP, MA and LP. We choose to solve it as a LP for the reasons explained above.

11.3 Enhanced parameter estimation and significance testing : RELAX and GLRT

From the SPICE power estimates (which are a function of both f and e) we calculate the maximum over e :

$$\bar{p}(f) = \max_e p(e, f) \quad (11.11)$$

where $p(e, f)$ denotes the SPICE power estimate p_l corresponding to a certain grid point l ($l = 2, \dots, Q+1$). Then, 10 peaks of $\bar{p}(f)$ (or, on occasions, fewer) are picked for further analysis (if there is prior information that more than 10 planets are orbiting the star in question we can of course consider more than 10 peaks). The frequency and eccentricity estimates corresponding to these 10 peaks are used as initial estimates for RELAX [5] [55], a parametric NLS estimation algorithm, that refines the SPICE estimates. The initial estimates for RELAX obtained via SPICE are, indeed, essential for RELAX to converge to the true parameters, else, RELAX initialized with arbitrary values might converge to a local minimum, which could lead to parameter estimates very much different from the true parameters. We refer the reader to [5] for the exact steps in RELAX algorithm, where we have applied RELAX, initialized with estimates obtained via a different approach, for parameter estimation of radial velocity data. Here, we would like to mention that the periastron passage times corresponding to the peaks chosen from SPICE estimates, which were not estimated till now, are estimated by RELAX. In RELAX, the initial

estimate of a periastron time corresponding to a peak with frequency estimate (\hat{f}) is taken to be $1/(2\hat{f})$.

The reasons for running RELAX on SPICE estimates are twofold : Firstly, if the grid (\mathcal{G}) used in SPICE is not fine enough to include the unknown true parameters, then we might end up with biased estimates; using RELAX we can refine the SPICE estimates. The second reason is that by employing RELAX we can exploit the nice statistical properties of its estimates and establish the statistical significance of the estimated parameters. For example, in the normal homogenous noise case, the RELAX estimates are optimal in the maximum likelihood sense, see e.g. [55]. We can then use the generalized likelihood ratio test (GLRT) to determine the statistical significance of the different planetary contributions to the radial velocity data under consideration. Specifically, we first apply RELAX on the largest peak of the SPICE spectrum in (11.11) and use GLRT to test the null hypothesis that there are no planets (or in other words, that the data is made of white noise) against the hypothesis that there is at least one exoplanet. If the test rejects the null hypothesis then we proceed by applying RELAX on the first two largest peaks and by testing the hypothesis that there is one exoplanet in the data against the hypothesis that there are at least two exoplanets. For instance, for the following hypotheses

H_0 : There are no planets

H_1 : There is at least one exoplanet with orbital frequency \hat{f}_1 , eccentricity \hat{e}_1 and periastron time \hat{T}_1

the corresponding log-likelihood (LL) values are given by:

$$\begin{aligned} \text{LL}(H_0) &= -\frac{N}{2} \ln \left(\sum_{n=1}^N [y(t_n)]^2 \right) + C, \\ \text{LL}(H_1) &= -\frac{N}{2} \ln \left(\sum_{n=1}^N [y(t_n) - \hat{\alpha}_1 \cos(\hat{v}(t_n)) - \hat{\beta}_1 \sin(\hat{v}(t_n))]^2 \right) + C \end{aligned} \quad (11.12)$$

where C denotes an additive constant whose exact expression is not important, $\hat{v}(t)$ is calculated from the RELAX estimates \hat{f}_1 , \hat{e}_1 and \hat{T}_1 , and $\hat{\alpha}_1$ and $\hat{\beta}_1$ are the RELAX estimates of α and β corresponding to $(\hat{e}_1, \hat{f}_1, \hat{T}_1)$. Then the GLRT is given by

$$\text{LLR} \stackrel{\Delta}{=} 2(\text{LL}(H_0) - \text{LL}(H_1)) \stackrel{H_1}{\underset{H_0}{\gtrless}} \Lambda \quad (11.13)$$

where Λ denotes a fixed threshold, which is usually chosen such that $\text{Prob}_{\chi_5^2}(\text{LLR} \leq \Lambda) = \xi$, where χ_5^2 denotes the chi-square distribution with 5 degrees of freedom, and ξ determines the significance level of the test. However, this will be valid only if the parameters of interest are identifiable under both the

Planet No.	Previous work				This work			
	e	f (cycles/day)	T (days)	K (m/s)	e (SPICE/RELAX)	f (cycles/day) (SPICE/RELAX)	T (days)	K (m/s)
1	0.06	0.0052	112.2	63.9	0.1/0.02	0.0055/0.0052	27.72	61.46
2	0.20	0.0333	9.1	46.6	0.2/0.13	0.0331/0.0333	30.06	45.11

Table 11.1. Parameters of the planets of HD 9446 star system.

hypotheses, which is not true in our case as can be clearly seen that the parameters of interest are unidentifiable under H_0 . To tackle this issue, we chose to determine the threshold via Monte-Carlo simulations. Given the sampling instants $\{t_n\}_{n=1}^N$, 10^4 realizations of white Gaussian noise sequences were generated and their corresponding LLRs were calculated. Finally, from the histogram of LLR, the value corresponding to a significance level of $\xi = 10^{-2}$ was chosen as the threshold (Λ) for the GLRT.

11.4 Real-life radial-velocity data processing

In this section, we consider the application of the algorithm introduced in the previous sections to real life radial velocity data. Our goal is to detect the exoplanets in a star system and estimate their orbital parameters in the Keplerian model. We will consider the radial velocity data set of HD 9446 : a class G type star with a planet in a circular orbit and another planet in an elliptical orbit [43].

The RV measurements from the star HD 9446 consists of 79 samples spanning 4 years. The grid parameters f_{\max} and e_{\max} were found to be 0.5 cycles/day and 1, respectively, from the spectral window (not shown here). After applying SPICE, RELAX and GLRT, two planets have been detected as shown in Fig 11.1. The threshold used in GLRT was calculated to be $\Lambda = 26.5$ from Monte-Carlo simulations. The estimated orbital parameters of the two detected planets are shown in Table 11.1 along with the results from [43]. It can be seen clearly from the table that the estimated orbital parameters of the two detected planets match well the previously reported results. Fig 11.2 also confirms that the fitted data (from both our work and the previous work) match reasonably well the observed data. The LL values corresponding to the two models in Table 11.1 are as follows :

$$\begin{aligned} \text{LL}_{\text{prev. work}} &= 136.65 \\ \text{LL}_{\text{this work}} &= 156.48 \end{aligned} \tag{11.14}$$

11.5 Conclusions

In this chapter we introduced an accurate and reliable technique of radial velocity data analysis for exoplanet detection. The SPICE method used here

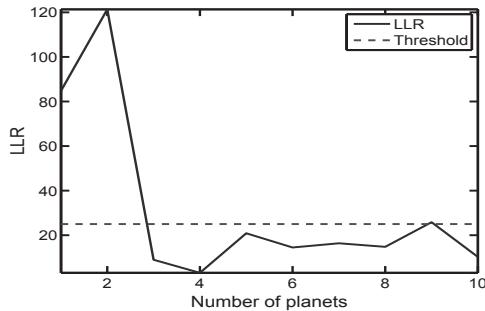


Figure 11.1. The LLR.

for computing an initial estimate of orbital parameters is globally convergent, does not need selection of any hyperparameters, and is computationally efficient. The SPICE estimates were then refined by RELAX and the statistical significance of the resultant estimates was determined by a GLRT. Finally, the tests conducted on a real-life data set showed that the results obtained by the proposed method fit the observed data slightly better than the previously reported Keplerian model (see, for instance, the LL values in (11.14)).

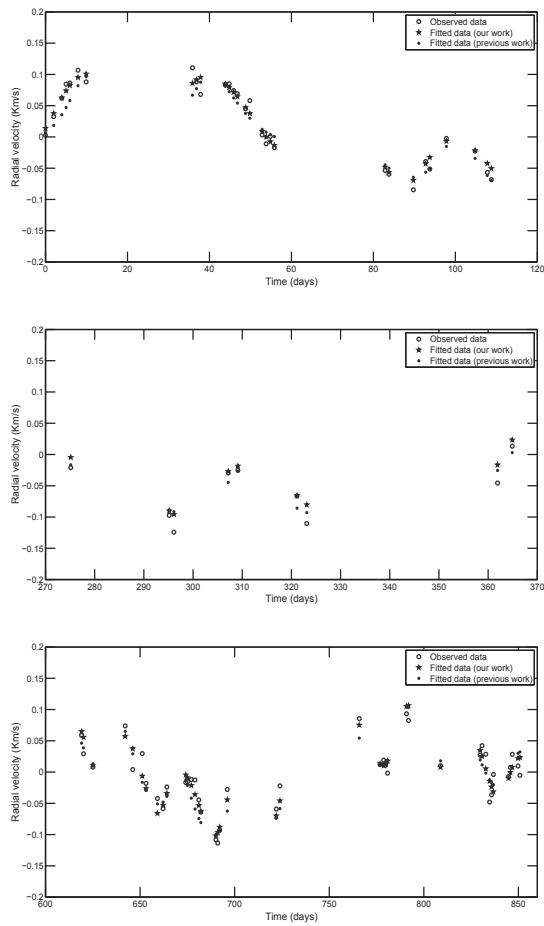


Figure 11.2. Comparison of the observed data and fitted data obtained from our work and the previous work.

12. Linear Systems, Sparse Solutions, and Sudoku

12.1 Introduction

Sudoku is a logic-based number-placement puzzle which is usually solved by hand for fun. For example, a 9×9 Sudoku puzzle may look as in fig. 12.1. The initial set of occupied cells are called the “clues” of the puzzle. The objective is to fill the empty cells in the puzzle such that the digits $1, \dots, 9$ occur only once in each row, each column and each 3×3 box. There are various computer algorithms available for solving Sudoku based on, e.g., backtracking, brute force search, stochastic search, set covering methods and constraint programming [116], [9]. Recently [71] proposed a Sinkhorn balancing algorithm to solve the puzzle.

In this chapter, we show that the puzzle ruleset can be expressed as an underdetermined linear system. We then prove that the Sudoku solution is the sparsest solution of this linear system and hence it can be obtained by l_0 norm minimization. However solving a l_0 norm minimization problem is hard in general, so we solve the much simpler problem of minimizing the l_1 norm and show numerically that this approach solves representative Sudoku puzzles. This result establishes a connection between the task of solving Sudoku and the emerging field of sparse signal representation [23],[127].

This chapter is organized as follows. Section II shows how the ruleset of the puzzle can be expressed as an underdetermined linear system. Section III proves the main result of the chapter that l_0 norm minimization solves Sudoku. In section IV, we show numerically that l_1 norm minimization solves representative Sudoku puzzles and briefly mention the conditions for equivalence of the solutions to the l_0 and l_1 norm minimization problems. Section V concludes the chapter.

12.2 Sudoku ruleset as a linear system

Let S denote an $N \times N$ Sudoku puzzle with a unique solution. As in [71], the content of cell n of S is denoted by $S_n \in \{1, 2, \dots, N\}$ for $n = 1, 2, \dots, N^2$, with the cells enumerated in row order. Let $i_n = [I(S_n = 1), I(S_n = 2), \dots, I(S_n = N)]^T$ denote the indicator vector associated with cell n , where $I(S_n = k)$ is the indicator function that is equal to one when $S_n = k$ and zero otherwise. Let x

	1	7	8	9				
3	8							
	9		5	6				
	9		7					
	3	1			2			
		4	5		8			
	5		6	2	4	9		
6	7	3	4	9	5	1		
4							3	

Figure 12.1. An example of a 9×9 Sudoku puzzle

denote a vector of size N^3 constructed as shown below:

$$x = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_{N^2} \end{bmatrix} \quad (12.1)$$

Taking the 9×9 Sudoku puzzle shown in fig. 12.1 as an example, the four types of constraints that one needs to satisfy to solve the puzzle are:

- Row constraints: Each row of S should comprise all digits $1, \dots, 9$.
- Column constraints: Each column of S should comprise all digits $1, \dots, 9$.
- Box constraints: Each 3×3 highlighted box of S should comprise all digits $1, \dots, 9$.
- Cell constraints: Each cell of S should be filled.

The cell constraints listed above are redundant as any of the other three types of constraints will ensure that all the cells of S are filled. However we retain them as they will be needed when we relax the puzzle problem in the next section. Each of the above constraints and the clues can be expressed as linear equality constraints on the elements of x . For example for the puzzle in fig. 12.1, the constraint that the first row of S should comprise all the digits $1, \dots, 9$ can be expressed as:

$$\left[I_{9 \times 9} \mid I_{9 \times 9} \mid \dots \mid I_{9 \times 9} \mid 0_{9 \times 648} \right] x = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (12.2)$$

where $I_{9 \times 9}$ denotes the 9×9 identity matrix and $0_{9 \times 648}$ denotes a matrix of size 9×648 with all elements equal to zero. Similarly the constraint that the first column of S should contain all digits $1, \dots, 9$ can be expressed as:

$$\left[I_{9 \times 9} \mid 0_{9 \times 72} \mid I_{9 \times 9} \mid 0_{9 \times 72} \mid \dots \mid I_{9 \times 9} \mid 0_{9 \times 72} \right] x = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (12.3)$$

The constraint that the 3×3 box in the top-left corner of S should contain all digits $1, \dots, 9$ can be expressed as:

$$\left[\begin{array}{c|c|c|c|c|c} J_{9 \times 27} & 0_{9 \times 54} & J_{9 \times 27} & 0_{9 \times 54} & J_{9 \times 27} & 0_{9 \times 540} \end{array} \right] x = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (12.4)$$

where

$$J_{9 \times 27} = \left[\begin{array}{c|c|c} I_{9 \times 9} & I_{9 \times 9} & I_{9 \times 9} \end{array} \right] \quad (12.5)$$

The constraint that the first cell S_1 should be filled can be expressed as:

$$1^T i_1 = 1, \quad 1 = [1 \ 1 \ \dots \ 1]^T \quad (12.6)$$

i.e.,

$$[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots \ \dots \ 0 \ 0]x = 1. \quad (12.7)$$

Finally the clues can also be expressed as linear equality constraints on x . For example, the clue that cell S_2 takes the value 1 can be expressed as:

$$i_2 = [1 \ 0 \ 0 \ 0 \ \dots \ 0]^T \quad (12.8)$$

i.e.,

$$[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \dots \ 0]x = 1. \quad (12.9)$$

By combining all these constraints, the linear equality constraints on x can be expressed in generic form as:

$$Ax = \begin{bmatrix} A_{row} \\ A_{col} \\ A_{box} \\ A_{cell} \\ A_{clue} \end{bmatrix} x = b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (12.10)$$

where A_{row} , A_{col} , A_{box} , A_{cell} and A_{clue} denote the matrices associated with the different constraints on the elements of x . For an $N \times N$ puzzle, the size of A is given by $(4N^2 + C) \times N^3$, where C denotes the number of clues. For the 9×9 puzzle in fig. 12.1, the size of A is 355×729 and hence the linear system (12.10) is underdetermined and has an infinite number of solutions. However, evidently, not all the solutions of (12.10) lead to a valid Sudoku solution. In fact for a puzzle with a unique solution there is only one solution x_S , with the elements of x_S made only of 0's and 1's. Hereafter we will call x_S the Sudoku solution.

12.3 l_0 norm minimization

As implied by the above discussion, the Sudoku solution x_S is sparse, that is it has few nonzero elements. To be more precise x_S is N^2 sparse (i.e. it has N^2 elements different from zero, out of N^3 elements). Consider the following optimization problem:

$$P_0 : \min_x \|x\|_0 \text{ s.t. } Ax = b \quad (12.11)$$

where $\|x\|_0$ represents the number of nonzero elements in x . Let x_{P_0} denote the solution of P_0 .

Proposition 1 *For a puzzle S with a unique solution, x_S , we have that $x_{P_0} = x_S$.*

Proof: If we show that x_S , the unique solution of S , is the sparsest solution of (12.10), then the theorem is proved. We know that $\|x_S\|_0 = N^2$. Let us assume that there exists an

$$x' = \begin{bmatrix} i'_1 \\ i'_2 \\ \vdots \\ i'_{N^2} \end{bmatrix}$$

such that $x' \neq x_S$ and $\|x'\|_0 < N^2$. This implies that at least one i'_j in x' should be equal to $[0, 0, \dots, 0]^T$. But this violates the cell constraint that every cell in S should be filled, i.e. for any j , $1^T i'_j = 1$, where $1 = [1, 1, \dots, 1]^T$. Thus x' cannot be a solution to (12.10); this means that all feasible solutions of (12.10)

must satisfy $\|x\|_0 = \sum_{j=1}^{N^2} \|i_j\|_0 \geq N^2$ where the equality holds only for $x = x_S$.

For any other $x \neq x_S$, there will be at least one i_j with $\|i_j\|_0 > 1$ and hence x_S is the unique sparsest solution of (12.10), i.e. $x_{P_0} = x_S$.

Finding the sparsest solution of (12.10) might not be a hard problem, as the solution has lot of structure that could be exploited in the search. However it is not obvious how to exploit this structure, and without doing so solving an l_0 norm minimization problem is hard in general. Therefore following [23] and [127] we relax the problem and instead solve a simpler l_1 norm minimization problem.

12.4 l_1 norm minimization

We determine x by solving the problem:

$$P_1 : \min_x \|x\|_1 \text{ s.t. } Ax = b \quad (12.12)$$

	A	B	C	D
No. of puzzles	10	10	20	20
No. of puzzles that Linpro solved	10	10	20	20
No. of puzzles that Sinkhorn solved	10	10	20	20
No. of puzzles that Bintprog solved	9	8	18	13

Table 12.1. Comparison of Linpro, Sinkhorn [71] and Bintprog [9] for $A = \text{"Easy"}$, $B = \text{"Medium"}$, $C = \text{"Hard"}$ and $D = \text{"Evil"}$ puzzles.

where $\|x\|_1 = \sum_{k=1}^{N^3} |x_k|$. Let x_{P_1} denote the solution of P_1 ; while it is not guaranteed that $x_{P_1} = x_{P_0}$, similarly to what happens in the sparse representation literature [23], [127], P_1 solves most Sudoku puzzles, that is for most puzzles P_1 finds the sparsest x that solves (12.10). The algorithm based on P_1 is called “Linpro”, as P_1 requires solving a linear program (which can be efficiently done by means of standard software packages (see e.g., [39])).

We have considered numerous Sudoku puzzles of varying levels of difficulty in which the puzzles and their designated level of difficulty are given in [45], [1]. In most cases Linpro gave the exact Sudoku solution. For example, Linpro successfully solved 200 puzzles of dimensions 9×9 and 16×16 from [45]. Table 12.1 compares the performance of Linpro, the Sinkhorn balancing algorithm (“Sinkhorn”) of [71] and the binary integer programming method (“Bintprog”) of [9] for puzzles of four different levels of difficulty. The puzzles which were used for testing the different methods in table 12.1 are chosen randomly from [1]. It is seen from table 12.1 that both Linpro and Sinkhorn have higher success rates than Bintprog.

Finally we comment on the equivalence between P_0 and P_1 . In the context of Sudoku, analyzing the conditions under which P_1 is equivalent to P_0 may shed light on the class of very hard Sudoku puzzles. For example, Linpro fails to solve the tricky puzzle shown in fig. 12.2 which is mentioned in [71] as one of the puzzles that cannot be solved by Sinkhorn. The Linpro algorithm successfully completes only 17 cells out of 55 empty cells of the puzzle; in other words Linpro fails to find the sparsest solution of the linear system associated with the puzzle. To tackle this problem, following [24], we tried using the iterative reweighted l_1 norm minimization algorithm outlined in table 12.2; this algorithm solved the puzzle by finding the sparsest solution of the underlying linear system. The Bintprog method also solved the tricky puzzle.

The presently known sufficient conditions on A for checking the equivalence of P_0 and P_1 , like the restricted isometry property (RIP) [23] and Kashin, Garnaev, Gluskin (KGG) inequality [127], do not hold for Sudoku. So analyzing the equivalence of l_0 and l_1 norm solutions in the context of Sudoku requires novel conditions for sparse recovery. Deriving such conditions is beyond the scope of this letter.

Initialization

$W^0 = I$, where I denotes the identity matrix of size $N^3 \times N^3$.

Iteration

At the i^{th} iteration, the vector of x , denoted by x^i , is obtained by solving the following problem:

$$\min_x \|W^{i-1}x\|_1 \text{ s.t. } Ax = b$$

$$\text{where } W_{k,k}^{i-1} = \frac{1}{|x_k^{i-1}| + \varepsilon}, \quad 0 < \varepsilon < 1.$$

The parameter ε can be tuned to obtain the sparsest solution of the linear system. We have used $\varepsilon = 0.5$.

Termination

The iteration is terminated when $\|x^i - x^{i-1}\|^2$ is less than 10^{-4} .

Table 12.2. Iterative reweighted l_1 norm minimization algorithm for solving Sudoku.

		3		9		8	1
			2			6	
5				1		7	
8	9						
		5	6		1	2	
						3	7
		9		2			8
7				4			
2	5	8			6		

Figure 12.2. An example of a tricky Sudoku puzzle

12.5 Conclusions

Inspired by [9] and [71], we showed that the Sudoku ruleset can be expressed as a linear system of equations. We then proved that seeking the minimum l_0 norm solution of this linear system will solve the puzzle. We then finally showed numerically that the simpler problem of minimizing the l_1 norm of the solution solves many representative Sudoku puzzles. Thus this chapter illustrates an interesting example of how l_1 norm minimization can be used to solve Sudoku.

Acknowledgements

I would like to thank Prof. Peter Stoica for providing me an opportunity to work with him. I would also like to thank the present and past members of Syscon for being helpful and supportive during my last five years of study. Finally, I would like to thank my mom and dad for all the support and strength they have given me.

References

- [1] <http://www.websudoku.com/>.
- [2] SAM-P8 : Compressed sensing and sparse signal representations. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 2011.
- [3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [4] A. Aldroubi and K. Gröchenig. Nonuniform sampling and reconstruction in shift-invariant spaces. *SIAM Review*, 43(4):585–620, 2001.
- [5] P. Babu, P. Stoica, J. Li, Z. Chen, and J. Ge. Analysis of radial velocity data by a novel adaptive approach. *The Astronomical Journal*, 139:783–793, February 2010.
- [6] Prabhu Babu and Petre Stoica. Spectral analysis of nonuniformly sampled data - a review. *Digital Signal Processing*, 20(2):359–378, 2010.
- [7] S. Baisch and G. H. R. Bokelmann. Spectral analysis with incomplete time series: an example from seismology. *Computers and Geosciences*, 25:739–750, 1999.
- [8] W.J. Bangs. *Array processing with generalized beam-formers*. PhD thesis, Yale University, New Haven, CT, 1971.
- [9] A.C. Bartlett and A.N. Langville. An Integer Programming Model for the Sudoku Problem. *Journal of Online Mathematics and its Applications*, 8, 2008.
- [10] A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.
- [11] L. Benedict, H. Nobash, and C. Tropea. Estimation of turbulent velocity spectra from laser Doppler data. *Meas. Sci. Technol.*, (11):1089–1104, 2000.
- [12] O. Besson and P. Stoica. Decoupled estimation of DOA and angular spread for a spatially distributed source. *IEEE Transactions on Signal Processing*, 48(7):1872–1882, 2000.
- [13] O. N. Bjornstad and W. Falck. Nonparametric spatial covariance functions: Estimation and testing. *Environmental and Ecological Statistics*, 8:53–70, 2001.
- [14] S. Bourguignon, H. Carfantan, and T. Böhm. SparSpec: a new method for fitting multiple sinusoids with irregularly sampled data. *Astronomy and Astrophysics*, 462(1):379–387, 2007.
- [15] S. Bourguignon, H. Carfantan, and J. Idier. A sparsity-based method for the estimation of spectral lines from irregularly sampled data. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):575–585, 2007.
- [16] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ Press, 2004.
- [17] P.J. Brockwell and R.A. Davis. *Time Series: Theory and Methods*. New York: Springer-Verlag, 2nd ed., 1991.

- [18] P. M. T. Broersen. Automatic spectral analysis with missing data. *Digital Signal Processing*, 16:754–766, 2006.
- [19] P. M. T. Broersen, S. de Waele, and R. Bos. Autoregressive spectral analysis when observations are missing. *Automatica*, 40:1495–1504, 2004.
- [20] P.M.T. Broersen. *Automatic Autocorrelation and Spectral Analysis*. Springer-Verlag New York Inc, 2006.
- [21] T.P. Bronez. Spectral estimation of irregularly sampled multidimensional processes by generalized prolate spheroidal sequences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(12):1862–1873, 1988.
- [22] N.R. Butt and A. Jakobsson. Coherence spectrum estimation from nonuniformly sampled sequences. *IEEE Signal Processing Letters*, 17(4):339–342, 2010.
- [23] EJ Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [24] E.J. Candes, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.
- [25] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.
- [26] A. Cumming. Detectability of extrasolar planets in radial velocity surveys. *Monthly Notices of the Royal Astronomical Society*, 354(4):1165–1176, 2004.
- [27] J.W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM Journal on Numerical Analysis*, pages 10–26, 1967.
- [28] PJ Daniell. On the theoretical specification and sampling properties of autocorrelated time-series. *J. Roy. Stat. Soc. Ser. B*, 8:88–90, 1946.
- [29] DL Donoho and Y. Tsaig. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.
- [30] J. H. Ender. On compressive sensing applied to radar. *Signal Processing*, 90(5):1402–1414, 2010.
- [31] L. Eyer and P. Bartholdi. Variable stars: Which Nyquist frequency? *Astronomy and Astrophysics, Supplement Series*, 135:1–3, February 1999.
- [32] H. G. Feichtinger and K. Gröchenig. "Theory and practice of irregular sampling", chapter in *Wavelets-Mathematics and Applications*, J. J. Benedetto and W. Frazier, eds, pages 305–363. CRC, Boca Raton, FL, 1993.
- [33] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numerische Mathematik*, 69(4):423–440, Feb 1995.
- [34] J.J. Fuchs. Convergence of a sparse representations algorithm applicable to real or complex data. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):598–605, 2007.
- [35] J. Ge, J. van Eyken, S. Mahadevan, C. DeWitt, S.R. Kane, R. Cohen, A. Vanden Heuvel, S.W. Fleming, P. Guo, G.W. Henry, et al. The First Extrasolar Planet Discovered with a New-Generation High-Throughput Doppler Instrument. *The Astrophysical Journal*, 648(1):683–695, 2006.

- [36] G Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.
- [37] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:1–26, 2003.
- [38] IF Gorodnitsky and BD Rao. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.
- [39] M. Grant, S. Boyd, and Y. Ye. CVX: Matlab software for disciplined convex programming. <http://stanford.edu/boyd/cvx>.
- [40] JR Guerci. *Space-Time Adaptive Processing for Radar*. Artech House Publishers, 2003.
- [41] P. Hall, N. Fisher, and B. Hoffmann. On the nonparametric estimation of covariance functions. *The Annals of Statistics*, 22(4):2115–2134, 1994.
- [42] W.K. Harteveld, R.F. Mudde, and H.E.A. Van den Akker. Estimation of turbulence power spectra for bubbly flows from laser Doppler anemometry signals. *Chem. Eng. Sci.*, (60):6160–6168, 2005.
- [43] G. Hébrard, X. Bonfils, D. Ségransan, C. Moutou, X. Delfosse, F. Bouchy, I. Boisse, L. Arnold, M. Desort, RF Diaz, et al. The SOPHIE search for northern extrasolar planets. *Astronomy and Astrophysics*, 513, 2010.
- [44] J. R. Higgins. *Sampling Theory in Fourier and Signal Analysis - Foundations*. Clarendon, Oxford, U.K, 1996.
- [45] M. Huckvale. *The Big Book of Su Doku*. Orion, 2005.
- [46] D.R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.
- [47] A. Jakobsson, S.R. Alty, and J. Benesty. Estimating and time-updating the 2-D coherence spectrum. *IEEE Transactions on Signal Processing*, 55(5):2350–2354, 2007.
- [48] A.T. Jessup, W.K. Melville, and W.C. Keller. Breaking waves affecting microwave backscatter 1. Detection and verification. *Journal of Geophysical Research*, 96:20547–20559, 1991.
- [49] R. H. Jones. Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, 22(3):389–395, Aug 1980.
- [50] R. H. Jones. Fitting a continuous time autoregression to discrete data. In D. F. Findley, editor, *Applied Time Series Analysis II*, pages 651–682. Academic, New York, 1981.
- [51] SM Kay and SL Marple Jr. Spectrum analysis - a modern perspective. *Proceedings of the IEEE*, 69(11):1380–1419, 1981.
- [52] E. Kreindler and A. Jameson. Conditions for nonnegativeness of partitioned matrices. *IEEE Transactions on Automatic Control*, 17(1):147–148, 1972.
- [53] E. K. Larsson. *Identification of Stochastic Continuous-Time Systems*. PhD thesis, Uppsala University, 2004.
- [54] H. Li, P. Stoica, and J. Li. Computationally efficient maximum likelihood estimation of structured covariance matrices. *IEEE Transactions on Signal Processing*, 47(5):1314–1323, 1999.
- [55] J. Li and P. Stoica. Efficient mixed-spectrum estimation with applications to target feature extraction. *IEEE Transactions on Signal Processing*,

- 44(2):281–295, 1996.
- [56] J. Li and P. Stoica, editors. *MIMO Radar Signal Processing*. J. Wiley & Sons, 2008.
- [57] A. W. C. Liew, J. Xian, S. Wu, D. Smith, and H. Yan. Spectral estimation in unevenly sampled space of periodically expressed microarray time series data. *BMC Bioinformatics*, 8:137–156, 2007.
- [58] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193–228, 1998.
- [59] N. R. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(1):10–33, July 1976.
- [60] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, 2008.
- [61] A. Maleki and D.L. Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):330–341, 2010.
- [62] D. Malioutov, M. Cetin, and AS Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022, 2005.
- [63] G.W. Marcy, R.P. Butler, D. Fischer, S.S. Vogt, J.J. Lissauer, and E.J. Rivera. A pair of resonant planets orbiting GJ 876. *The Astrophysical Journal*, 556(1):296–301, 2001.
- [64] G.W. Marcy, R.P. Butler, S.S. Vogt, D. Fischer, and J.J. Lissauer. A planetary companion to a nearby M4 dwarf, Gliese 876. *The Astrophysical Journal*, 505:147–150, 1998.
- [65] R. J. Martin. *Irregularly Sampled Signals: Theories and Techniques for Analysis*. PhD thesis, University College London, 1998.
- [66] R. J. Martin. Autoregression and irregular sampling: Spectral estimation. *Signal Processing*, 77:139–157, 1999.
- [67] F. Marvasti. *Nonuniform Sampling, Theory and Practice*. Kluwer, Norwell, MA, 2001.
- [68] E. Masry. Poisson sampling and spectral estimation of continuous time processes. *IEEE Transactions on Information Theory*, 24(2):173–183, 1978.
- [69] A. Mathias, F. Grond, R. Guardans, D. Seese, M. Canela, H.H. Diebner, and G. Baiocchi. Algorithms for spectral analysis of irregularly sampled time series. *Journal of Statistical Software*, 11(2):1–30, 2004.
- [70] M. Mishali and Y. C. Eldar. Blind multi-band signal reconstruction: Compressed sensing of analog signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009, March 2009.
- [71] T. K. Moon, J. H. Gunther, and J. Kupin. Sinkhorn solves Sudoku. *IEEE Transactions on Information Theory*, pages 1741–1746, 2009.
- [72] C. Moutou, M. Mayor, F. Bouchy, C. Lovis, F. Pepe, D. Queloz, NC Santos, S. Udry, W. Benz, G. Lo Curto, et al. The HARPS search for southern extra-solar planets: IV. Three close-in planets around HD 2638, HD 27894 and HD 63454. *Astronomy & Astrophysics*, 439(1):367–373, 2005.
- [73] Y. Nesterov. Squared functional systems and optimization problems. *High performance optimization*, 13:405–440, 2000.

- [74] SJ O'Toole, CG Tinney, HRA Jones, RP Butler, GW Marcy, B. Carter, and J. Bailey. Selection functions in doppler planet searches. *Monthly Notices of the Royal Astronomical Society*, 392(2):641–654, 2009.
- [75] B. Ottersten, P. Stoica, and R. Roy. Covariance matching estimation techniques for array signal processing applications. *Digital Signal Processing*, 8(3):185–210, 1998.
- [76] D.B. Percival and A.T. Walden. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge University Press, 1993.
- [77] J.R. Petit, J. Jouzel, D. Raynaud, NI Barkov, JM Barnola, I. Basile, M. Bender, J. Chappellaz, M. Davis, G. Delaygue, et al. Climate and atmospheric history of the past 420,000 years from the Vostok ice core, Antarctica. *Nature*, 399(6735):429–436, 1999.
- [78] M.B. Priestley. *Spectral Analysis and Time Series*. Academic Press : London, UK, 1981.
- [79] L. Pronzato. Optimal experimental design and some related control problems. *Automatica*, 44(2):303–325, 2008.
- [80] S. Provencher. Parameters Estimation of Complex Multitone Signal in the DFT Domain. *IEEE Transactions on Signal Processing*, 59(7):3001–3012, 2011.
- [81] E.J. Rivera, J.J. Lissauer, R.P. Butler, G.W. Marcy, S.S. Vogt, D.A. Fischer, T.M. Brown, G. Laughlin, and G.W. Henry. A 7.5 M planet orbiting the nearby star, GJ 876. *The Astrophysical Journal*, 634(1):625, 2005.
- [82] D. H. Roberts, J. Lehar, and J. W. Dreher. Time series analysis with CLEAN. I. Derivation of a spectrum. *The Astronomical Journal*, 93(4):968–989, April 1987.
- [83] J.B. Roberts. Spectral analysis of signals sampled at random times. In N. Jones, editor, *Digital Signal Processing*. Stevenage, UK : Peter Peregrinus, 1982.
- [84] W. Roberts, H. He, J. Li, and P. Stoica. Probing waveform synthesis and receiver filter design. *IEEE Signal Processing Magazine*, 27(4):99–112, 2010.
- [85] W. Roberts, P. Stoica, J. Li, T. Yardibi, and F.A. Sadjadi. Iterative adaptive approaches to MIMO radar imaging. *IEEE Journal of Selected Topics in Signal Processing*, 4(1):5–20, 2010.
- [86] J. D. Scargle. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *Astrophysical Journal*, 263:835–853, December 1982.
- [87] M. Schulz and K. Stattegger. Spectrum: Spectral analysis of unevenly spaced paleoclimatic time series. *Computers and Geosciences*, 23(9):929–945, 1997.
- [88] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [89] P. Stoica and P. Babu. Algebraic derivation of Elfving theorem on optimal experiment design and some connections with sparse estimation. *IEEE Signal Processing Letters*, 17:743–745, 2010.
- [90] P. Stoica and P. Babu. Maximum-likelihood non-parametric estimation of smooth spectra from irregularly sampled data. *IEEE Transactions on Signal Processing*, 60:5746–5758, 2012.

- [91] P. Stoica and P. Babu. Sparse estimation of spectral lines : Grid selection problems and their solutions. *IEEE Transactions on Signal Processing*, 60:962–967, 2012.
- [92] P. Stoica and P. Babu. SPICE and LIKES : Two hyperparameter-freemethods for sparse-parameter estimation. *Signal Processing*, 92:1580–1590, 2012.
- [93] P. Stoica, P. Babu, and J. Li. New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sampled data. *IEEE Transactions on Signal Processing*, 59(1):35–47, 2011.
- [94] P. Stoica, E.G. Larsson, and J. Li. Adaptive filter-bank approach to restoration and spectral analysis of gapped data. *The Astronomical Journal*, 120(4):2163–2173, 2000.
- [95] P. Stoica, H. Li, and J. Li. Amplitude estimation of sinusoidal signals: Survey, new results, and an application. *IEEE Transactions on Signal Processing*, 48(2):338–352, 2000.
- [96] P. Stoica, J. Li, and H. He. Spectral analysis of nonuniformly sampled data: a new approach versus the periodogram. *IEEE Transactions on Signal Processing*, 57(3):843–858, 2009.
- [97] P. Stoica, J. Li, and J. Ling. Missing data recovery via a nonparametric iterative adaptive approach. *IEEE Signal Processing Letters*, 16(4):241–244, 2009.
- [98] P. Stoica and R. Moses. *Spectral Analysis of Signals*. Prentice Hall, Upper Saddle River, N.J., 2005.
- [99] P. Stoica and A. Nehorai. MUSIC, maximum likelihood, and Cramer-Rao bound. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(5):720–741, 1989.
- [100] P. Stoica and N. Sandgren. Spectral analysis of irregularly-sampled data: Paralleling the regularly-sampled data approaches. *Digital Signal Processing*, 16(6):712–734, October 2006.
- [101] P. Stoica and Y. Selen. Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher. *IEEE Signal Processing Magazine*, 21(1):112–114, 2004.
- [102] P. Stoica and Y. Selen. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.
- [103] P. Stoica and T. Sundin. On nonparametric spectral estimation. *Circuits, Systems, and Signal Processing*, 18(2):169–181, 1999.
- [104] J.F. Sturm. Using SeDuMi 1. 02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1):625–653, 1999.
- [105] X. Tan, W. Roberts, J. Li, and P. Stoica. MIMO radar imaging via SLIM. *IEEE Transactions on Signal Processing*, submitted 2009.
- [106] A. Tarczynski and N. Allay. Spectral analysis of randomly sampled signals: Suppression of aliasing and sampler jitter. *IEEE Transactions on Signal Processing*, 52(12):3324–3334, Dec 2004.
- [107] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [108] CG Tinney, R.P. Butler, G.W. Marcy, H.R.A. Jones, A.J. Penny, C. McCarthy, B.D. Carter, and D.A. Fischer. Three Low-Mass Planets from the Anglo-Australian Planet Search 1. *The Astrophysical Journal*, 623(2):1171–1179, 2005.

- [109] M.E. Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.
- [110] R. Touzi, A. Lopes, J. Bruniquel, and P.W. Vachon. Coherence estimation for SAR imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 37(1):135–149, 1999.
- [111] C. Tropea. Laser Doppler anemometry: Recent developments and future challenges. *Measurements Science and Technology*, 6:605–619, 1995.
- [112] R. Venkataramani and Y. Bresler. Perfect reconstruction formulas and bounds on aliasing error in sub-Nyquist nonuniform sampling of multiband signals. *IEEE Transactions on Information Theory*, 46(6):2173–2183, Sep 2000.
- [113] R. Wallin, AJ Isaksson, and L. Ljung. An iterative method for identification of ARX models from incomplete data. In *Proceedings of the 39th IEEE conference on Decision and Control*. Sydney, Australia, 2000.
- [114] Y. Wang, P. Stoica, J. Li, and T. L. Marzetta. Nonparametric spectral analysis with missing data via the EM algorithm. *Digital Signal Processing*, 15:192–206, March 2005.
- [115] P. Welch. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.
- [116] Wikipedia. Algorithmics of Sudoku — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 13-May-2009].
- [117] D. Wipf and S. Nagarajan. A new view of automatic relevance determination. *Advances in neural information processing systems*, 20:1625–1632, 2008.
- [118] D.P. Wipf and B.D. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004.
- [119] D.P. Wipf and B.D. Rao. An empirical Bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Transactions on Signal Processing*, 55(7):3704–3716, 2007.
- [120] J. Wright and Y. Ma. Dense error correction via ℓ_1 minimization. *IEEE Transactions on Information Theory*, 56(7):3540–3560, 2010.
- [121] C. Yang and G. Wei. A Noniterative Frequency Estimator With Rational Combination of Three Spectrum Lines. *IEEE Transactions on Signal Processing*, 59(10):5065–5070, 2011.
- [122] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Bagherer. Iterative adaptive approach for sparse signal representation with sensing applications. *IEEE Transactions on Aerospace and Electronic Systems*, 2009, to appear.
- [123] L. Yu, H. Sun, J. P. Barbot, and G. Zheng. Bayesian compressive sensing for cluster structured sparse signals. *Signal Processing*, 92(1):259–269, 2012.
- [124] S. Yu, A. S. Khwaja, and J. Ma. Compressed sensing of complex-valued data. *Signal Processing*, 92(2):357–362, 2012.
- [125] Y. Yu. Monotonic convergence of a general algorithm for computing optimal designs. *Annals of Statistics*, 38(3):1593–1606, 2010.
- [126] Zechmeister, M. and Kurster, M. The generalised Lomb-Scargle periodogram. *Astronomy and Astrophysics*, 496(2):577–584, 2009.
- [127] Y. Zhang. A Simple proof for recoverability of L1-minimization: go over or under? *Rice University CAAM Technical Report TR05-09*, 2005.

- [128] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

