SPICE and LIKES : Two hyperparameter-free methods for sparse-parameter estimation

Petre Stoica, Prabhu Babu[*]

Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden

Abstract

SPICE (**SP**arse **I**terative **C**ovariance-based **E**stimation) is a recently introduced method for sparse-parameter estimation in linear models using a robust covariance fitting criterion that does not depend on any hyperparameters. In this paper we revisit the derivation of SPICE to streamline it and to provide further insights into this method. LIKES (**LIK**elihood-based **E**stimation of **S**parse parameters) is a new method obtained in a hyperparameter-free manner from the maximum-likelihood principle applied to the same estimation problem as considered by SPICE. Both SPICE and LIKES are shown to provide accurate parameter estimates even from scarce data samples, with LIKES being more accurate than SPICE at the cost of an increased computational burden.

Key words: Scarce data, sparse parameter estimation methods, robust covariance fitting, maximum-likelihood method, SDP, SOCP, spectral analysis, range-Doppler imaging.

[*] This work was supported in part by the Swedish Research Council (VR) and the European Research Council (ERC). Please address all the correspondence to Prabhu Babu, Phone: (46) 18-471-3394; Fax: (46) 18-511925; Email: prabhu.babu@it.uu.se

1. Introduction and problem formulation

Consider the following linear model:

$$\begin{aligned} \boldsymbol{y} &= \sum_{k=1}^{M} \boldsymbol{a}_k x_k + \boldsymbol{e} \\ &= [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_M \; \boldsymbol{I}] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{e} \end{bmatrix} = \boldsymbol{B}\boldsymbol{\beta} \end{aligned} \tag{1}$$

where

$$\begin{aligned} \boldsymbol{x} &= [x_1, \cdots, x_M]^T \\ \boldsymbol{\beta} &= \left[\boldsymbol{x}^T \; \boldsymbol{e}^T\right]^T \\ \boldsymbol{B} &= [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_M \; \boldsymbol{I}] \triangleq [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{M+N}] \,. \end{aligned} \tag{2}$$

Furthermore, in (1) $\boldsymbol{y} \in \mathbb{C}^N$ denotes the observation vector, $\{\boldsymbol{a}_k \in \mathbb{C}^N\}_{k=1}^M$ is a set of given vectors, $\{x_k \in \mathbb{C}\}_{k=1}^M$ are unknown parameters, and $\boldsymbol{e} \in \mathbb{C}^N$ is a noise term; the matrix $\boldsymbol{B} \in \mathbb{C}^{N \times (M+N)}$ and the vector $\boldsymbol{\beta} \in \mathbb{C}^{M+N}$, introduced in (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., [1] [2] [3] [4] [5] [6] [7] [8] and the many references there) can be reduced to the estimation of $\boldsymbol{x}$ in the above linear model with $M >> N$ (i.e. scarce data) and with only a few elements of $\boldsymbol{x}$ different from zero (i.e., a sparse parameter vector). Note that if the noise vector $\boldsymbol{e}$ in (1) were assumed to be sparse then the parameter vector $\boldsymbol{x}$ could be exactly recovered from $\boldsymbol{y}$, under fairly weak conditions, for instance by the methods proposed later on in the paper (see, e.g., [9]). However, we do *not* make this assumption here and therefore our problem is the estimation, rather than the exact recovery, of $\boldsymbol{x}$.

There are several methods in the literature that can be used to estimate the sparse parameter vector in (1). However, most of these methods require the selection of one or more user parameters (aka hyperparameters), which is usually a daunting task (see, e.g., the lucid discussion in [5]). In [1] [2] we have recently introduced a **SP**arse **I**terative **C**ovariance-based **E**stimation (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 [1] and [2] was on temporal and, respectively, spatial spectral analysis, and owing to this focusing the estimation of $\boldsymbol{x}$ in (1) was less stressed than that of other (related) parameters, see Section 2 for details. Here we change the emphasis and treat $\boldsymbol{x}$ as the parameter vector of primary interest, which is normally the case.

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 [10] [11], to derive a novel method for estimating $\boldsymbol{x}$ in (1), which we designate by the acronym LIKES (**LIK**elihood-based **E**stimation of **S**parse parameters). The concise derivation of LIKES, presented in Section 3, relies on that of SPICE in Section 2; indeed, we exploit the links between LIKES and SPICE to simplify the description of the steps of LIKES.

In Section 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. **I**terative **R**eweighted ℓ_1-norm minimization (IRL1) [12] [13] and **B**asis **P**ursuit (BP) [14].

2. SPICE

We will make the working assumption that the elements of $\boldsymbol{\beta}$ 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 $\boldsymbol{y}$ is given by:

$$\boldsymbol{R} = E(\boldsymbol{y}\boldsymbol{y}^*) = \boldsymbol{B}\boldsymbol{P}\boldsymbol{B}^* \tag{3}$$

where the superscript $*$ denotes the conjugate transpose, and

$$\boldsymbol{P} = \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & p_M & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \sigma_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \sigma_N \end{bmatrix} \triangleq \begin{bmatrix} p_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & p_M & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & p_{M+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & p_{M+N} \end{bmatrix}. \tag{4}$$

The SPICE estimation metric is the following weighted covariance fitting criterion (see [1] [2] [15] and also the references therein) :

$$\left\| \boldsymbol{R}^{-1/2}(\boldsymbol{R} - \boldsymbol{y}\boldsymbol{y}^*) \right\|^2 \tag{5}$$

where $\|\cdot\|$ denotes the Frobenius norm for matrices (as well as the Euclidean norm for vectors), and $\boldsymbol{R}^{-1/2}$ is a Hermitian square-root of the inverse matrix $\boldsymbol{R}^{-1}$ (which is assumed to exist). The following comments on (3) and (5) are in order:

- The type of model (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., [1] [2] [16]).

- 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 3) can be readily modified to take this information into account : see [1] [2] for SPICE ; the modification of LIKES is similar. However, our experience is that imposing this condition on $\{\sigma_k\}$, even when it holds, does not improve the estimation accuracy significantly. Consequently, for the sake of conciseness, we will omit any discussion on imposing it and refer instead to the cited references for details on this aspect.

- We make a similar remark on the case of multiple data vectors (aka snapshots), in which the vectors $\boldsymbol{y}$ and $\boldsymbol{x}$ in (1) should be replaced by matrices $\boldsymbol{Y}$ and $\boldsymbol{X}$ with $\boldsymbol{X}$ being row-wise sparse. We refer the interested reader to [2] for details on the extension of SPICE to this case; the extension of LIKES is similar.

- We will estimate the power vector

$$\boldsymbol{p} = [p_1, \cdots, p_{M+N}]^T \quad (p_k \geq 0) \tag{6}$$

by minimizing the covariance fitting criterion in (5). In some cases, such as in the spectral analysis applications considered in [1] [2], estimating $\boldsymbol{p}$ 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., [1]). With these facts in mind, here we put the emphasis on estimating $\boldsymbol{x}$ as the parameter vector of main interest, somewhat in contrast to what has been done in [1] [2]. While we still consider the minimization of (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 $\boldsymbol{\beta}$ that led to the observed data vector $\boldsymbol{y}$ has the following maximum aposteriori (MAP)-like form (e.g., [11]) :

$$\boldsymbol{\beta} = \boldsymbol{P}\boldsymbol{B}^*\boldsymbol{R}^{-1}\boldsymbol{y} \tag{7}$$

see below for details.

Returning to (5), a simple calculation shows that this fitting criterion can be re-written as:

$$\begin{aligned}&\operatorname{tr}\left[(\boldsymbol{I} - \boldsymbol{y}\boldsymbol{y}^*\boldsymbol{R}^{-1})(\boldsymbol{R} - \boldsymbol{y}\boldsymbol{y}^*)\right]\\ &= \operatorname{tr}(\boldsymbol{R}) + \|\boldsymbol{y}\|^2\boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y} - 2\|\boldsymbol{y}\|^2\end{aligned} \tag{8}$$

where

$$\operatorname{tr}(\boldsymbol{R}) = \sum_{k=1}^{M+N} p_k\|\boldsymbol{b}_k\|^2. \tag{9}$$

It follows from (8) and (9) that the minimization problem of interest here is the following:

$$\min_{\boldsymbol{p}} \boldsymbol{y}^* \boldsymbol{R}^{-1} \boldsymbol{y} + \sum_{k=1}^{M+N} w_k^2 p_k \tag{10}$$

where the weights

$$w_k = \tfrac{\|\boldsymbol{b}_k\|}{\|\boldsymbol{y}\|} \tag{11}$$

do not depend on $\boldsymbol{p}$. At this point we remark on the fact that in [1] [2] the second term in (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 (10) as it stands. The benefit of doing so is twofold : i) the SPICE algorithm obtained from (10) is slightly simpler than the version in [1]; and ii) the $\{p_k\}$ obtained from (10) are related to $\{|x_k|\}$ via a known scaling factor (see (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 [1] [2], the said scaling factor depends on unknown quantities.

The minimization problem in (10) is *convex*. Indeed, it can be cast as the following semi-definite program (SDP) [17] (with α being an auxiliary variable):

$$\begin{aligned} &\min_{\boldsymbol{p},\alpha} \alpha + \sum_{k=1}^{M+N} w_k^2 p_k \\ &\text{s.t.} \begin{bmatrix} \alpha & \boldsymbol{y}^* \\ \boldsymbol{y} & \boldsymbol{R} \end{bmatrix} \geq 0. \end{aligned} \tag{12}$$

However we do not recommend obtaining the solution to (10) by solving the above SDP. The reason is that the currently available SDP solvers for (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 (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, (10) can also be cast as a second-order cone program (SOCP), which can be solved much more efficiently than the SDP in (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 [2], but it was not shown explicitly for (10). In the next sub-section, we provide a simple proof of the SOCP reformulation of (10).

2.1. SOCP-based solver

Consider the following augmented problem:

$$\begin{aligned} &\min_{\boldsymbol{p},\boldsymbol{\beta}} \boldsymbol{\beta}^* \boldsymbol{P}^{-1} \boldsymbol{\beta} + \sum_{k=1}^{M+N} w_k^2 p_k \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y}. \end{aligned} \tag{13}$$

The use of the symbol $\boldsymbol{\beta}$ to denote the extra variables in (13) is not accidental : indeed, the constraint in (13) is nothing but the data equation (1). Similarly to (10), the problem (13) can also be shown to be *convex*. Note that if some $\{p_k\}$ are equal to zero, then $\boldsymbol{P}^{-1}$ in (13) should be replaced by the pseudo-inverse of $\boldsymbol{P}$; the following analysis carries over the latter case with only relatively minor modifications.

The property of (13), which is important here, is that the minimization wrt $\boldsymbol{\beta}$ yields the original problem. Specifically :

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^* \boldsymbol{P}^{-1} \boldsymbol{\beta} = \boldsymbol{y}^* \boldsymbol{R}^{-1} \boldsymbol{y} \quad \text{s.t. } \boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y}. \tag{14}$$

Consequently, *the solutions* $\boldsymbol{p}$ *to (10) and (13) are identical.* To prove (14), we need to show that

$$\boldsymbol{\beta}^* \boldsymbol{P}^{-1} \boldsymbol{\beta} \geq \boldsymbol{y}^* \boldsymbol{R}^{-1} \boldsymbol{y} \quad \text{s.t. } \boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y} \tag{15}$$

or equivalently (making use of the constraint) :

$$\boldsymbol{\beta}^* \boldsymbol{P}^{-1} \boldsymbol{\beta} \geq \boldsymbol{\beta}^* \boldsymbol{B}^* \boldsymbol{R}^{-1} \boldsymbol{B} \boldsymbol{\beta} \tag{16}$$

which holds if and only if

$$\boldsymbol{P}^{-1} \geq \boldsymbol{B}^* \boldsymbol{R}^{-1} \boldsymbol{B} \tag{17}$$

(for two Hermitian matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the notation $\boldsymbol{A} \geq \boldsymbol{B}$ means that the difference matrix $\boldsymbol{A} - \boldsymbol{B}$ is positive semi-definite). By a standard property of partitioned positive semi-definite matrices ([18]), equation (17) is equivalent to:

$$\begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{B}^* \\ \boldsymbol{B} & \boldsymbol{R} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^{-1/2} \\ \boldsymbol{B}\boldsymbol{P}^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^{-1/2} & \boldsymbol{P}^{1/2}\boldsymbol{B}^* \end{bmatrix} \geq 0 \tag{18}$$

which is obviously true. The result (15) is therefore proved. Furthermore, it can be easily verified that the minimizing vector $\boldsymbol{\beta}$ is given by:

$$\boldsymbol{\beta} = \boldsymbol{P}\boldsymbol{B}^* \boldsymbol{R}^{-1} \boldsymbol{y} \tag{19}$$

a fact that will be used in the next sub-section (note that (19) is the MAP-like estimate of $\boldsymbol{\beta}$ in (7), which was mentioned in the previous discussion).

Making use of the observation that the solution $\boldsymbol{p}$ to (10) is identical to the minimizer $\boldsymbol{p}$ of (13), we will obtain this solution from (13); interestingly the latter problem can be solved more efficiently than (10), as explained in the following, in spite of the additional variable $\boldsymbol{\beta}$ in (13).

The minimization of (13) wrt $\boldsymbol{p}$, for fixed $\boldsymbol{\beta}$, decouples in $(M+N)$ one-dimensional problems with the following generic form :

$$\min_{p\geq 0} \tfrac{|\beta|^2}{p} + w^2 p \tag{20}$$

or, equivalently,

$$\min_{p\geq 0} \left(\tfrac{|\beta|}{\sqrt{p}} - w\sqrt{p}\right)^2 + 2w|\beta|. \tag{21}$$

It follows easily from (21) that the minimizer $\boldsymbol{p}$ of (13), for fixed $\boldsymbol{\beta}$, is:

$$p_k = \tfrac{|\beta_k|}{w_k} \quad k = 1, \ldots, M+N \tag{22}$$

and also that the minimization problem wrt $\boldsymbol{\beta}$ that remains to be solved is given by

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{k=1}^{M+N} w_k |\beta_k| \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y}. \end{aligned} \tag{23}$$

This problem can be readily cast as an SOCP [17] (with $\{\alpha_k\}$ being auxiliary variables):

$$\begin{aligned} &\min_{\{\alpha_k\},\boldsymbol{\beta}} \sum_{k=1}^{M+N} w_k \alpha_k \\ &\text{s.t. } |\beta_k| \leq \alpha_k \quad k = 1, \ldots, M+N \\ &\boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y} \end{aligned} \tag{24}$$

which can be solved much more efficiently than the SDP in (12).

In summary, the SOCP-based SPICE algorithm consists of solving the SOCP in (24) to obtain an estimate of $\boldsymbol{\beta}$ (and hence of $\boldsymbol{x}$). The solution $\boldsymbol{p}$ to the original SPICE problem in (10), if desired, can be obtained from $\boldsymbol{\beta}$ via (22).

It is worth noting that in the case of *real-valued data*, the SOCP in (24) reduces to the following *linear program* (LP) (below, $\alpha_k, \beta_k \in \mathbb{R}$):

$$\begin{aligned}&\min_{\{\alpha_k,\beta_k\}} \sum_{k=1}^{M+N} w_k \alpha_k \\ &\text{s.t.} -\alpha_k \leq \beta_k \leq \alpha_k \; ; \; \alpha_k \geq 0 \; ; \quad k = 1, \ldots, M+N \\ &\boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{y}\end{aligned} \tag{25}$$

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 (25), such as homotopy-based methods and iterative thresholding-based algorithms (see, e.g., [19]).

For real-valued data, the use of such an efficient algorithm to solve the LP in (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 (13) than the SOCP-based algorithm of this section.

2.2. CA-based solver

The minimizers of the criterion in (13) wrt $\boldsymbol{\beta}$, for fixed $\boldsymbol{p}$, and wrt $\boldsymbol{p}$, for fixed $\boldsymbol{\beta}$, have been derived in the previous sub-section : they are given by the closed-form expressions in (19) and, respectively, (22). Therefore, the main ingredients of a CA for the minimization of (13) are already available. Let the super-index i indicate the i-th iteration, and let $\boldsymbol{P}(i)$ denote the matrix $\boldsymbol{P}$ 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) :

$$\begin{aligned}&p_k^i = |\beta_k^i|/w_k \\ &\beta_k^{i+1} = p_k^i \boldsymbol{b}_k^* \boldsymbol{R}^{-1}(i)\boldsymbol{y} \; ; \; \boldsymbol{R}(i) = \boldsymbol{B}\boldsymbol{P}(i)\boldsymbol{B}^*\end{aligned} \tag{26}$$

$(k = 1, \ldots, M{+}N; i = 0, 1, 2, \ldots)$. This iterative algorithm *converges globally* to the solution of (13) from any initial values $\{p_k > 0\}$ or $\{\beta_k \neq 0\}$ (see [1] and the references there). However, needless to say, the rate of convergence may depend on the initial value. Here we will initialize (26) with the element-wise least-squares estimate of $\boldsymbol{\beta}$ in (1) :

$$\beta_k^0 = \frac{\boldsymbol{b}_k^* \boldsymbol{y}}{\|\boldsymbol{b}_k\|^2} \quad k = 1, \ldots, M + N. \tag{27}$$

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 $\boldsymbol{A}$ of that problem).

3. LIKES

We will now make the additional assumption that $\boldsymbol{x}$ and $\boldsymbol{e}$ are circularly Gaussian distributed. This means that $\boldsymbol{y}$ has a circular Gaussian distribution with zero mean and covariance matrix equal to $\boldsymbol{R}$. Consequently the negative log-likelihood function associated with $\boldsymbol{y}$ is given (to within an additive constant) by:

$$f(\boldsymbol{p}) = \ln|\boldsymbol{R}| + \boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y}. \tag{28}$$

We will obtain an estimate of $\boldsymbol{p}$ by minimizing this function, i.e. :

$$\min_{\boldsymbol{p}} \boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y} + \ln|\boldsymbol{R}|. \tag{29}$$

The objective in (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 paper, (29) may be deemed to be statistically a more appealing estimation criterion than the covariance fitting metric in (10).

With regard to the form of the two fitting criteria in (10) and (29), we see that they share a common term, viz. $\boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y}$, which is a convex function of $\boldsymbol{p}$. The problem is that the second term in (29), unlike that in (10), is not convex. In fact this term can be shown to be a *concave function of* $\boldsymbol{p}$, see Appendix A. The consequence of this fact is, as shown in Appendix B, that (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 (29), which *decreases the criterion $f(\boldsymbol{p})$ 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 $\tilde{\boldsymbol{p}}$ be an arbitrary point in the parameter space, and let $\tilde{\boldsymbol{R}}$ 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 $\boldsymbol{p}$:

$$\begin{aligned} \ln|\boldsymbol{R}| &\leq \ln|\tilde{\boldsymbol{R}}| + \sum_{k=1}^{M+N} \operatorname{tr}\left(\tilde{\boldsymbol{R}}^{-1}\boldsymbol{b}_k\boldsymbol{b}_k^*\right)(p_k - \tilde{p}_k) \\ &= \ln|\tilde{\boldsymbol{R}}| - N + \operatorname{tr}(\tilde{\boldsymbol{R}}^{-1}\boldsymbol{R}) = \ln|\tilde{\boldsymbol{R}}| - N + \sum_{k=1}^{M+N} \tilde{w}_k^2 p_k \end{aligned} \tag{30}$$

where

$$\tilde{w}_k^2 = \boldsymbol{b}_k^*\tilde{\boldsymbol{R}}^{-1}\boldsymbol{b}_k. \tag{31}$$

It follows from (30) that :

$$f(\boldsymbol{p}) \leq \left(\ln|\tilde{\boldsymbol{R}}| - N\right) + \boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y} + \sum_{k=1}^{M+N} \tilde{w}_k^2 p_k \triangleq g(\boldsymbol{p}) \tag{32}$$

for any vectors $\tilde{\boldsymbol{p}}$ and $\boldsymbol{p}$. Note also that

$$f(\tilde{\boldsymbol{p}}) = g(\tilde{\boldsymbol{p}}). \tag{33}$$

The important implication of (32) and (33) is that we can decrease the function $f(\boldsymbol{p})$ from $f(\tilde{\boldsymbol{p}})$ to, let us say, $f(\hat{\boldsymbol{p}})$ by choosing $\hat{\boldsymbol{p}}$ as the minimum point of $g(\boldsymbol{p})$ or at least such that $g(\tilde{\boldsymbol{p}}) > g(\hat{\boldsymbol{p}})$:

$$f(\hat{\boldsymbol{p}}) \leq g(\hat{\boldsymbol{p}}) < g(\tilde{\boldsymbol{p}}) = f(\tilde{\boldsymbol{p}}) \tag{34}$$

(the first inequality above follows from (32), the second inequality from the definition of $\hat{\boldsymbol{p}}$, and the equality from (33).) This is in fact the underlying principle of the minimization-majorization approach to solving a given minimization problem, see, e.g. [20] [21]. The usefulness of the said approach depends on whether the minimization (or the decrease) of $g(\boldsymbol{p})$ is easier than that of $f(\boldsymbol{p})$. Here this is definitely true, as $g(\boldsymbol{p})$ is (to within a constant) a SPICE-like convex criterion function, compare it with (10). Consequently, the SPICE solvers of the previous section can be used to find a vector $\hat{\boldsymbol{p}}$ with the above property, for any given $\tilde{\boldsymbol{p}}$.

The procedure for the minimization of the negative log-likelihood function in (29), outlined above, is designated by the acronym LIKES (**LIK**elihood-based **E**stimation of **S**parse parameters). LIKES consists of an initialization stage (we can use the same initial values as for SPICE, see (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 (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 $\tilde{\boldsymbol{p}}$, build $\tilde{\boldsymbol{R}}$ and employ either of the SPICE solvers to obtain the next estimate $\hat{\boldsymbol{p}}$ (as well as $\hat{\boldsymbol{\beta}}$, if desired). Note that an off-the-shelf SOCP solver will compute $\hat{\boldsymbol{p}}$ as the minimum point of $g(\boldsymbol{p})$; on the other hand, if desired, the CA-based solver can be iterated only as long as it decreases $g(\boldsymbol{p})$ "significantly" and therefore it can be stopped before complete convergence to speed up the computation of $\hat{\boldsymbol{p}}$; the trade-off being that a presumably larger number of outer iterations may then be required.

Outer step. Set $\tilde{\boldsymbol{p}} = \hat{\boldsymbol{p}}$, and go to the inner step.

It is somewhat interesting to note that neither of the estimation methods presented in this paper 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 $\boldsymbol{p}$-dependent SPICE criterion in (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 $\boldsymbol{\beta}$-

dependent SPICE criterion in (23) where the objective function itself is the weighted ℓ_1-norm of $\boldsymbol{\beta}$. These observations also apply to LIKES whose inner step is a SPICE-like estimation problem.

We expect that the LIKES estimates of $\boldsymbol{\beta}$ and $\boldsymbol{p}$ 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 [16]) 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).

We end this section with a discussion on the relationship between LIKES and the **S**parse **B**ayesian **L**earning (SBL) approach of [10] and [11]. While both LIKES and SBL make use of the maximum-likelihood principle, as in (29), the two algorithms are quite different from one another : SBL is an **E**xpectation-**M**aximization (EM) - type of method, whereas LIKES is a **M**inimization-**M**ajorization (MM) - based technique (as explained above). In general EM algorithms are known to converge more slowly than MM counterparts (see, e.g., [21]) 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 [11] [10] (a detailed comparison of LIKES and SBL is left to future work). From a theoretical viewpoint, on the other hand, the analysis in [11] and [10] has a number of interesting results that also apply to LIKES *mutatis mutandis.* In particular, the cited papers (especially [11]) contain a detailed explanation of the sparsity of the LIKES/SBL parameter estimates : in a nutshell, if $\{p_k\}_{k=1}^{M}$ in (3) are assumed to have a certain prior then the resultant hierarchical prior for $\boldsymbol{x}$ can be shown to have a sparsity-inducing character. Furthermore, [11] includes a useful analysis of the global and local minima of the negative log-likelihood function in (28) which, in the case of local minima, holds under either noisy or noiseless conditions.

4. Numerical illustrations and concluding remarks

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 :

$$y(t_n) = \sum_{k=1}^{3} c_k e^{i\omega_k t_n} + e(t_n) \quad ; n = 1, \ldots, 50 \tag{35}$$

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 (35) are the following ones :

$$\begin{aligned} &c_1 = 5 \;\; c_2 = 5 \;\; c_3 = 10 \\ &\omega_1 = 2\pi\, 0.3 \;\; \omega_2 = 2\pi\, 0.5 \;\; \omega_3 = 2\pi\, 0.52. \end{aligned} \tag{36}$$

The signal-to-noise ratio (SNR) for (35)-(36) is defined as :

$$\text{SNR} = 10 \log(100/\sigma). \tag{37}$$

Let

$$\boldsymbol{a}_p = \begin{bmatrix} e^{i\nu_p t_1} \\ \vdots \\ e^{i\nu_p t_N} \end{bmatrix} \quad ; \, p = 1, \ldots, M \tag{38}$$

where

$$\nu_p = \tfrac{2\pi}{M} p \;\; ; \;\; N = 50 \;\; ; \;\; M = 1000. \tag{39}$$

Using the above notation we can re-write (35) in the form (1), viz.

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \tag{40}$$

with the parameter vector $\boldsymbol{x}$ having only three non-zero elements which are equal to $\{c_k\}_{k=1}^{3}$.

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 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 $\boldsymbol{y}$ and this is not an easy task. In order to estimate σ as reliably as possible we assume that *we know the sparsity of* $\boldsymbol{x}$ in (40) (i.e. the number of non-zero elements of $\boldsymbol{x}$). 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 (35) is three we can apply a powerful parametric method to $\boldsymbol{y}$ to estimate the signal parameters, see, e.g. [16]).

To estimate σ we make use of the initial estimate of $\boldsymbol{x}$ in (27), viz. $|x_k| = |\boldsymbol{a}_k^* \boldsymbol{y}|/\|\boldsymbol{a}_k\|^2$ (for $k = 1, \cdots, M$), 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 (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 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 2 we show one randomly selected plot from each of Figures 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 inserts 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 $\boldsymbol{x}$ 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 3 for 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, 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 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.

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. [22]):

$$\boldsymbol{y} = \sum_{k=-N+1}^{N-1} \sum_{l=-L}^{L-1} x_{kl} \boldsymbol{a}_{kl} + \boldsymbol{e}. \tag{41}$$

In (41), x_{kl} is the reflectivity coefficient for the cell corresponding to the k-th range bin and the l-th Doppler bin, $\boldsymbol{e}$ is a white noise with zero mean and variance σ, and

$$\boldsymbol{a}_{kl} = \boldsymbol{J}_k \begin{bmatrix} s_1 e^{i\omega_l} \\ \vdots \\ s_N e^{iN\omega_l} \end{bmatrix} \tag{42}$$

where ω_l is the frequency associated with the l-th Doppler bin,

$$\omega_l = \tfrac{\pi}{L} l, \tag{43}$$

and $\boldsymbol{J}_k$ denotes the following shifting matrix

$$\boldsymbol{J}_k = \left[\begin{array}{ccccc} \multicolumn{3}{c}{\overbrace{0 \quad \cdots \quad 1}^{k+1}} & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{array}\right] = \boldsymbol{J}^*_{-k} \quad k = 0, \ldots, N-1. \tag{44}$$

We consider the case of five targets present in (41), with the following parameters:

$$\begin{array}{rccccc} \text{Range bin } k: & 0 & 0 & 0 & 2 & -2 \\ \text{Doppler bin } l: & -15 & 5 & 10 & -15 & 20 \\ \text{Reflectivity coefficient } x_{kl}: & 5 & 5 & 10 & 5 & 5 \end{array} \tag{45}$$

Similarly to (37), we define the SNR as

$$\text{SNR} = 10 \log(100/\sigma). \tag{46}$$

We will show results for a data realization with SNR = 20dB, $N = 50$, and $L = 25$.

Note that the data vector $\boldsymbol{y}$ in (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 $\boldsymbol{y}$ as $|k|$ increases and therefore their estimation from $\boldsymbol{y}$ in (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 (45). Nevertheless, we will not estimate $\{x_{kl}\}$ only for these three values of k, but will use the data vector $\boldsymbol{y}$ to estimate *all* $\{x_{kl}\}$.

Evidently the data model (41) is already in the form (1) (with $M + N = 5000$) and hence we can directly apply the initial estimator in (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 $\{|x_{kl}|\}$, aka the estimated range-Doppler images, along with the true image are shown in Figure 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 4 only for $k \in [-20, 20]$.

Appendix A : Concavity proof

We prove here that $\ln|\boldsymbol{R}|$ is a concave function of $\boldsymbol{p}$ by showing that its Hessian matrix is negative semi-definite at any point in the parameter space. We have that :

$$\tfrac{\partial \ln|\boldsymbol{R}|}{\partial p_k} = \operatorname{tr}\left(\boldsymbol{R}^{-1}\tfrac{\partial \boldsymbol{R}}{\partial p_k}\right) = \boldsymbol{b}_k^* \boldsymbol{R}^{-1} \boldsymbol{b}_k \tag{47}$$

and

$$\tfrac{\partial^2 \ln|\boldsymbol{R}|}{\partial p_k \partial p_s} = -\boldsymbol{b}_k^* \boldsymbol{R}^{-1} \boldsymbol{b}_s \boldsymbol{b}_s^* \boldsymbol{R}^{-1} \boldsymbol{b}_k \triangleq -H_{ks}. \tag{48}$$

The matrix $\boldsymbol{H}$, introduced above, must therefore be shown to be positive semi-definite. In other words we have to prove that $\boldsymbol{g}^* \boldsymbol{H} \boldsymbol{g} \geq 0$ for any vector $\boldsymbol{g} = [g_1, \cdots, g_{M+N}]^T$. Let

$$\boldsymbol{X} = \sum_{k=1}^{M+N} g_k \boldsymbol{b}_k \boldsymbol{b}_k^* \tag{49}$$

and let $\operatorname{vec}(\boldsymbol{X})$ denote the vector made from the columns of $\boldsymbol{X}$ stacked on top of each other. Then a simple calculation shows that:

$$\begin{aligned} \boldsymbol{g}^* \boldsymbol{H} \boldsymbol{g} &= \sum_{p=1}^{M+N} \sum_{s=1}^{M+N} g_p^* g_s \operatorname{tr}\left(\boldsymbol{R}^{-1} \boldsymbol{b}_p \boldsymbol{b}_p^* \boldsymbol{R}^{-1} \boldsymbol{b}_s \boldsymbol{b}_s^*\right) = \operatorname{tr}\left(\boldsymbol{R}^{-1} \boldsymbol{X}^* \boldsymbol{R}^{-1} \boldsymbol{X}\right) \\ &= \operatorname{vec}^*(\boldsymbol{X}) \left(\boldsymbol{R}^{-T} \otimes \boldsymbol{R}^{-1}\right) \operatorname{vec}(\boldsymbol{X}) \end{aligned} \tag{50}$$

where $\otimes$ denotes the Kronecker matrix product. Because the matrix $\boldsymbol{R}^{-T} \otimes \boldsymbol{R}^{-1}$ is positive definite, it follows from (50) that $\boldsymbol{g}^*\boldsymbol{H}\boldsymbol{g} \geq 0$ and the proof is concluded.

Appendix B : Non-convexity proof

The Hessian matrix associated with the function $f(\boldsymbol{p})$ in (28) has the following elements :

$$\begin{aligned} \tfrac{\partial^2 f(\boldsymbol{p})}{\partial p_k \partial p_s} &= -\boldsymbol{b}_k^* \boldsymbol{R}^{-1} \boldsymbol{b}_s \boldsymbol{b}_s^* \boldsymbol{R}^{-1} \boldsymbol{b}_k + 2\mathrm{Re}(\boldsymbol{y}^* \boldsymbol{R}^{-1} \boldsymbol{b}_k \boldsymbol{b}_k^* \boldsymbol{R}^{-1} \boldsymbol{b}_s \boldsymbol{b}_s^* \boldsymbol{R}^{-1} \boldsymbol{y}) \\ &\stackrel{\Delta}{=} -H_{ks} + G_{ks} \quad k, s = 1, \ldots, M+N \end{aligned} \tag{51}$$

where $\mathrm{Re}(z)$ denotes the real part of z. The first term in (51) follows from (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 $\boldsymbol{y}^*\boldsymbol{R}^{-1}\boldsymbol{y}$, can be similarly shown to be positive semi-definite : $\boldsymbol{G} \geq 0$. For sufficiently small values of $\|\boldsymbol{p}\|$, $\boldsymbol{G}$ dominates $\boldsymbol{H}$ in (51); whereas the opposite is true if $\|\boldsymbol{p}\|$ is large enough. It follows that the Hessian matrix in (51) becomes positive semi-definite as $\|\boldsymbol{p}\|$ approaches zero, comes to be negative semi-definite as $\|\boldsymbol{p}\|$ approaches infinity, and in general is indefinite for some intermediate values of $\|\boldsymbol{p}\|$. With this observation, the proof is concluded.

References

[1] P. Stoica, P. Babu, 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) (2011) 35–47.

[2] P. Stoica, P. Babu, J. Li, SPICE: A sparse covariance based estimation method for array processing, IEEE Transactions on Signal Processing 59 (2) (2011) 629–638.

[3] M. Lustig, D. Donoho, J. Santos, J. Pauly, Compressed sensing MRI, IEEE Signal Processing Magazine 25 (2) (2008) 72–82.

[4] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1) (1996) 267–288.

[5] A. Maleki, D. Donoho, Optimally tuned iterative reconstruction algorithms for compressed sensing, IEEE Journal of Selected Topics in Signal Processing 4 (2) (2010) 330–341.

[6] S. Yu, A. S. Khwaja, J. Ma, Compressed sensing of complex-valued data, Signal Processing (2011) In press.

[7] L. Yu, H. Sun, J. P. Barbot, G. Zheng, Bayesian compressive sensing for cluster structured sparse signals, Signal Processing 92 (1) (2012) 259–269.

[8] J. H. Ender, On compressive sensing applied to radar, Signal Processing 90 (5) (2010) 1402–1414.

[9] J. Wright, Y. Ma, Dense error correction via ℓ_1 minimization, IEEE Transactions on Information Theory 56 (7) (2010) 3540–3560.

[10] M. Tipping, Sparse Bayesian learning and the relevance vector machine, The Journal of Machine Learning Research 1 (2001) 211–244.

[11] D. Wipf, B. Rao, Sparse Bayesian learning for basis selection, IEEE Transactions on Signal Processing 52 (8) (2004) 2153–2164.

[12] E. Candes, M. Wakin, S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, Journal of Fourier Analysis and Applications 14 (5) (2008) 877–905.

[13] H. Zou, The adaptive lasso and its oracle properties, Journal of the American Statistical Association 101 (476) (2006) 1418–1429.

[14] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing 20 (1) (1999) 33–61.

[15] B. Ottersten, P. Stoica, R. Roy, Covariance matching estimation techniques for array signal processing applications, Digital Signal Processing 8 (3) (1998) 185–210.

[16] P. Stoica, R. Moses, Spectral Analysis of Signals, Prentice Hall, Upper Saddle River, NJ, 2005.

[17] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ Press, 2004.

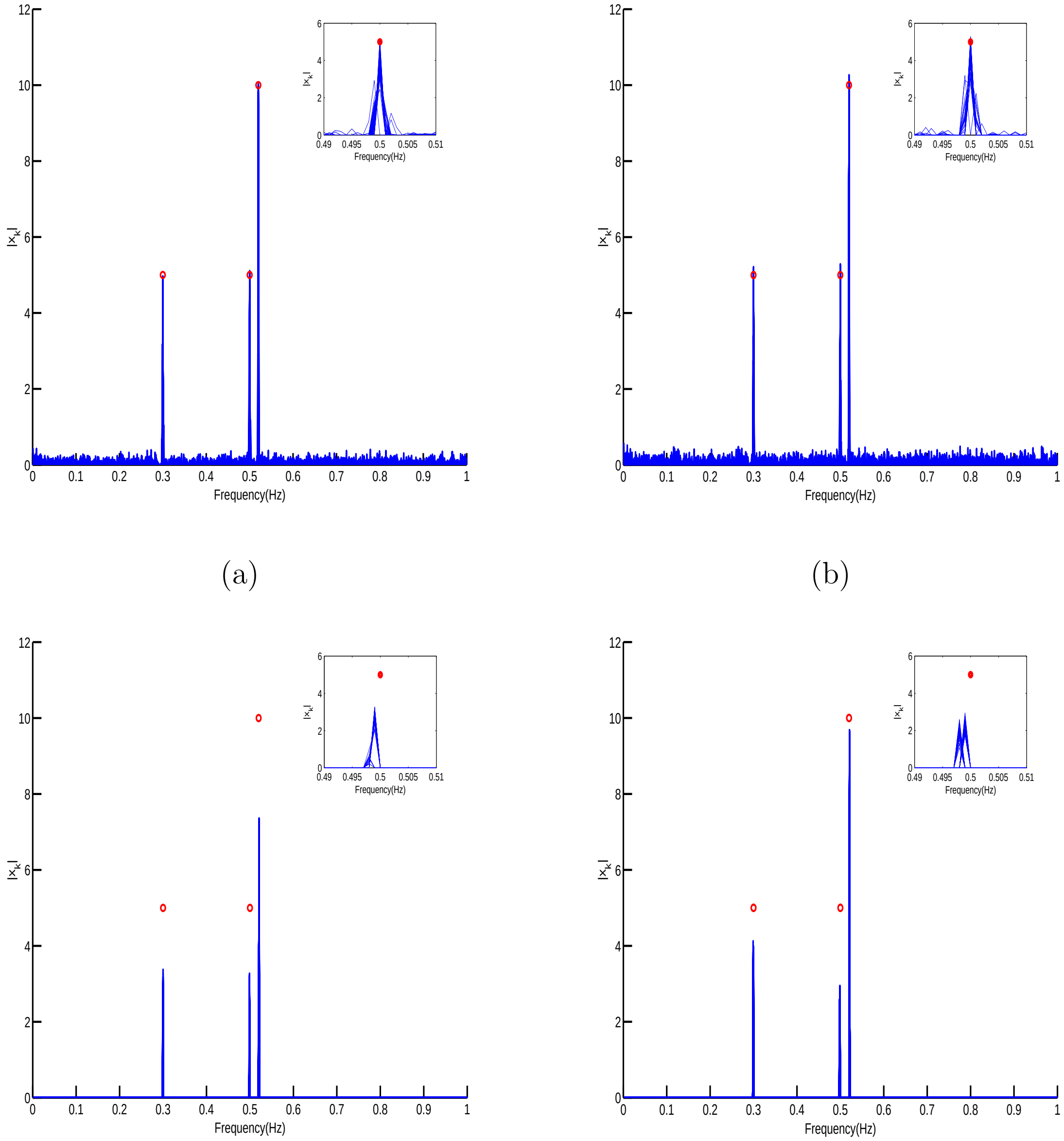
[18] E. Kreindler, A. Jameson, Conditions for nonnegativeness of partitioned matrices, IEEE Transactions on Automatic Control 17 (1) (1972) 147–148.

[19] D. Donoho, Y. Tsaig, Fast solution of ℓ_1-norm minimization problems when the solution may be sparse, IEEE Transactions on Information Theory 54 (11) (2008) 4789–4812.

[20] P. Stoica, Y. Selen, Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher, IEEE Signal Processing Magazine 21 (1) (2004) 112–114.

[21] D. Hunter, K. Lange, A tutorial on MM algorithms, The American Statistician 58 (1) (2004) 30–37.

[22] W. Roberts, H. He, J. Li, P. Stoica, Probing waveform synthesis and receiver filter design, IEEE Signal Processing Magazine 27 (4) (2010) 99–112.



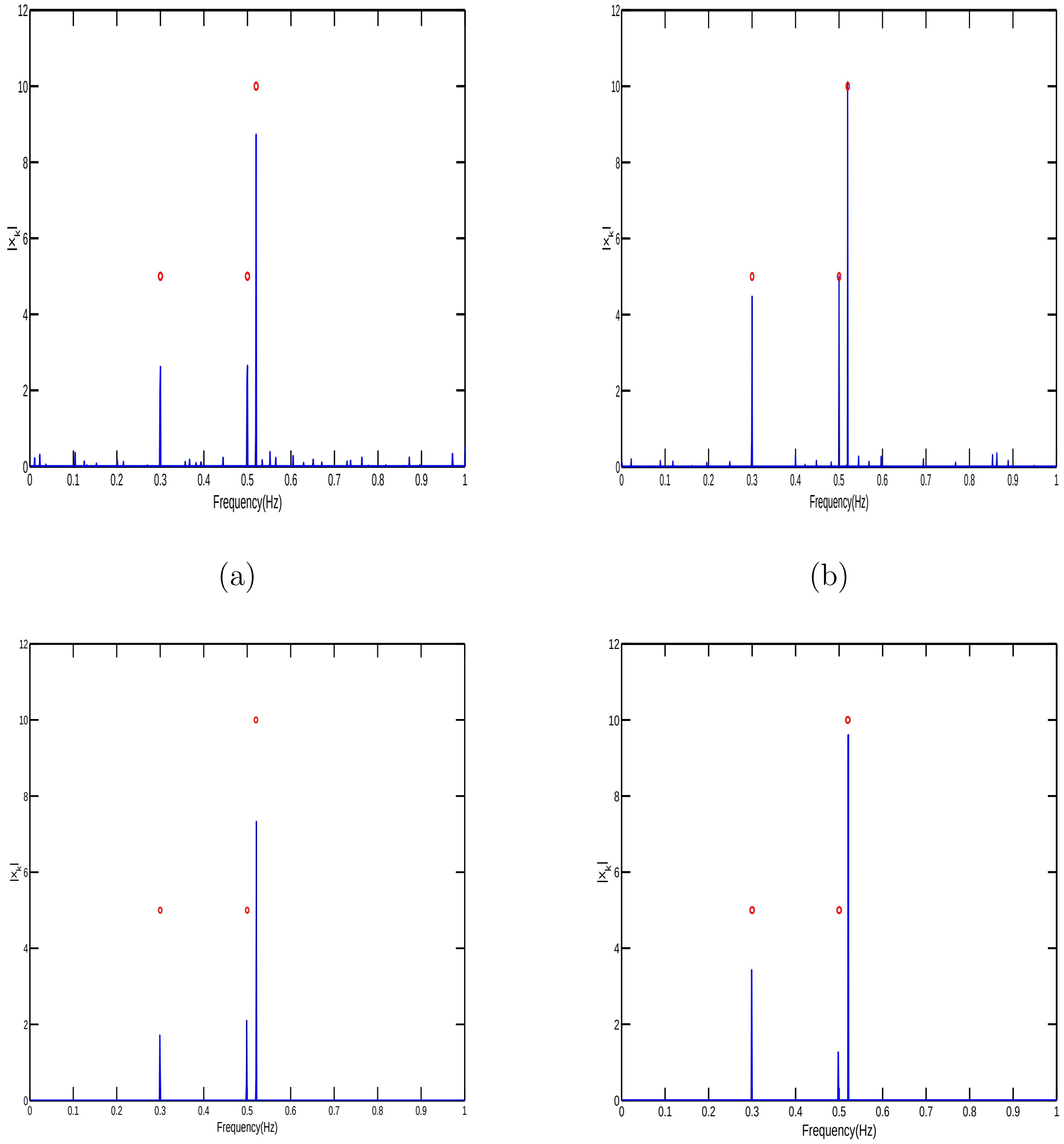
(a)

(b)

(c)

(d)

Figure 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



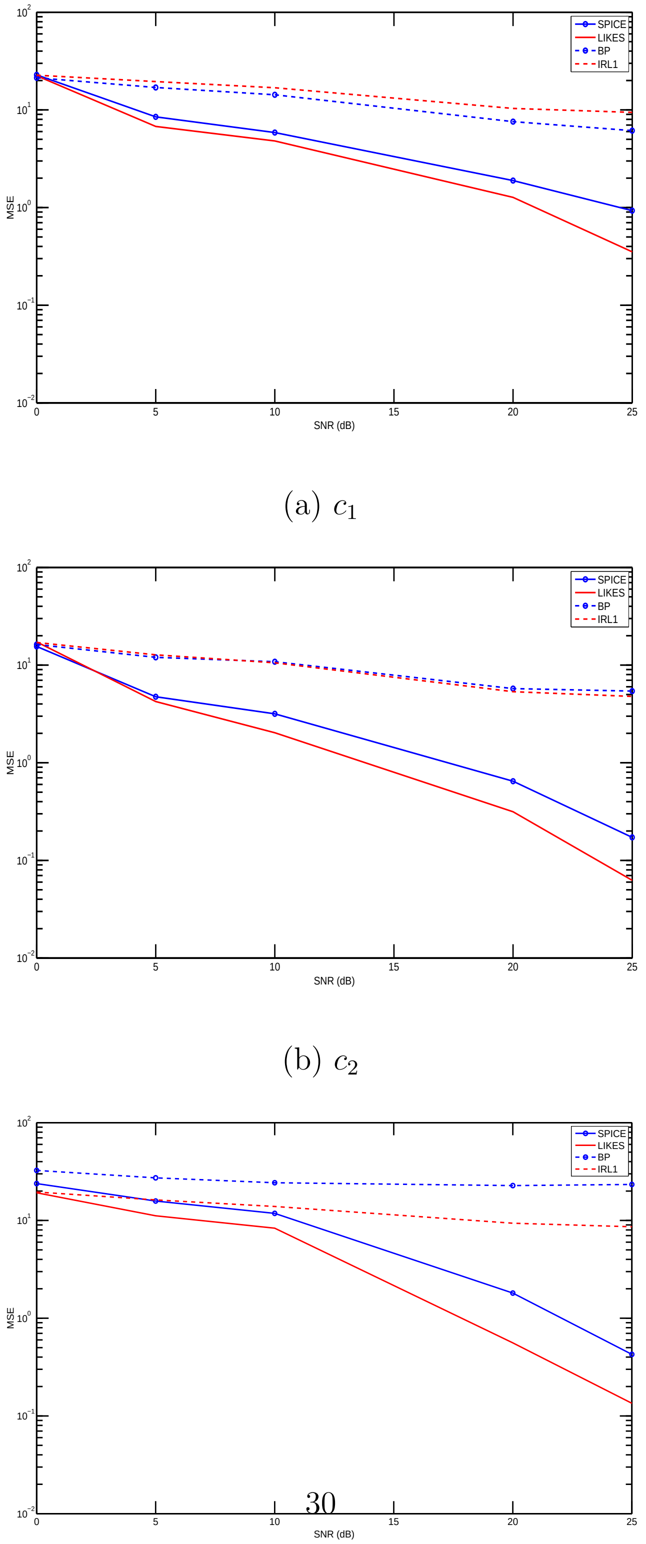
(a)

(b)

(c)

(d)

Figure 2: Four randomly selected plots of $\{|x_k|\}$ from Figure 1 for (a) SPICE (b) LIKES (c) BP and (d) IRL1.

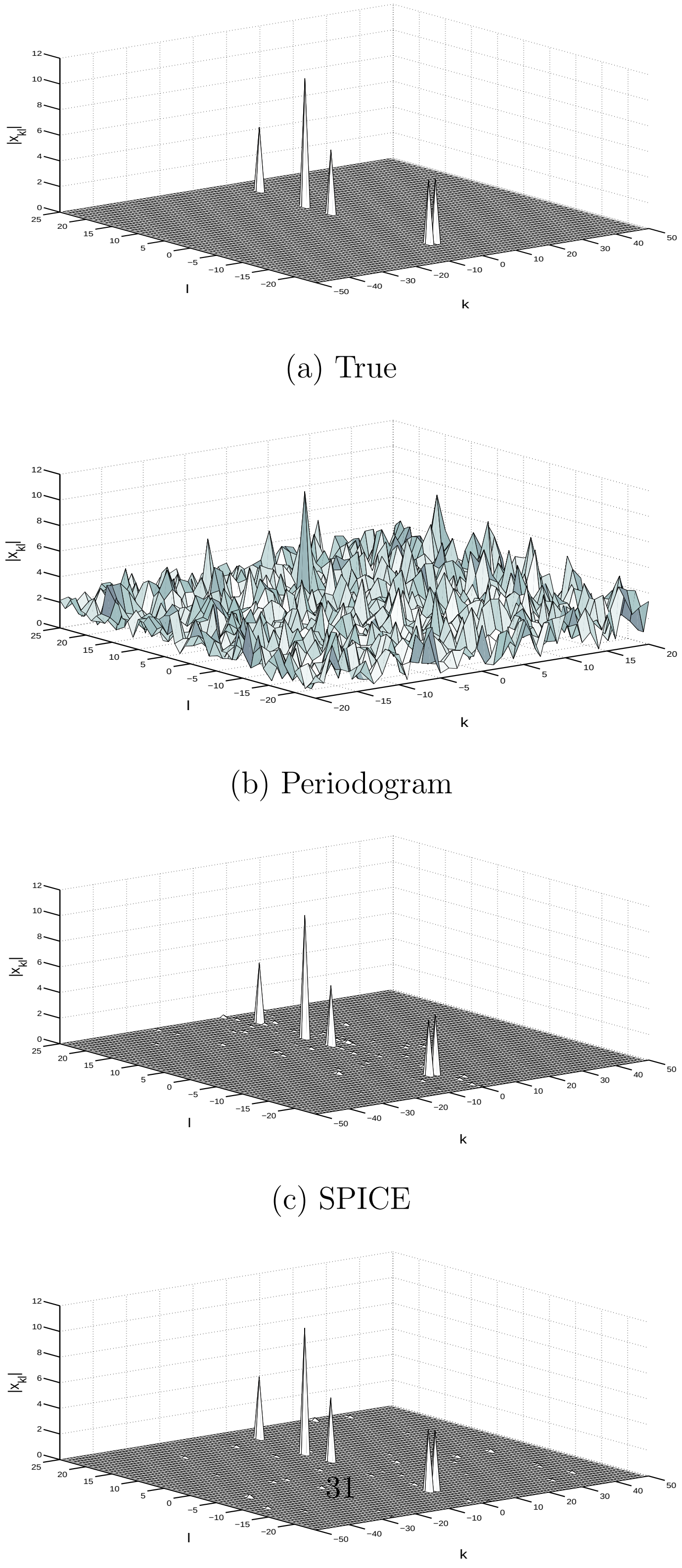


(a) c_1

(b) c_2

(c) c_3

Figure 3: MSE vs SNR for the SPICE, LIKES, BP and IRL1 estimates of $\{c_k\}_{k=1}^{3}$.



(a) True

(b) Periodogram

(c) SPICE

(d) LIKES

Figure 4: The true and estimated range-Doppler images.