Superfast Line Spectral Estimation

Hansen, Thomas Lundgaard; Fleury, Bernard Henri; Rao, Bhaskar D.

Published in:
I E E E Transactions on Signal Processing

DOI (link to publication from Publisher):
10.1109/TSP.2018.2807417

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Superfast Line Spectral Estimation

Thomas L. Hansen, Bernard H. Fleury, *Senior Member, IEEE*, and Bhaskar D. Rao, *Fellow, IEEE*

***Abstract*—A number of recent works have proposed to solve the line spectral estimation problem by applying off-the-grid extensions of sparse estimation techniques. These methods are preferable over classical line spectral estimation algorithms because they inherently estimate the model order. However, they all have computation times which grow at least cubically in the problem size, thus limiting their practical applicability in cases with large dimensions. To alleviate this issue, we propose a low-complexity method for line spectral estimation, which also draws on ideas from sparse estimation. Our method is based on a Bayesian view of the problem. The signal covariance matrix is shown to have Toeplitz structure, allowing superfast Toeplitz inversion to be used. We demonstrate that our method achieves estimation accuracy at least as good as current methods and that it does so while being orders of magnitudes faster.**

I. Introduction

The problem of line spectral estimation (LSE) has received significant attention in the research community for at least 40 years. The reason is that many fundamental problems in signal processing can be recast as LSE; examples include direction of arrival estimation using sensor arrays [1], [2], bearing and range estimation in synthetic aperture radar [3], channel estimation in wireless communications [4] and simulation of atomic systems in molecular dynamics [5].

In trying to solve the LSE problem, classical approaches include subspace methods [6] such as MUSIC [7] or ESPRIT [8] which estimate the frequencies based on an estimate of the signal covariance matrix. These approaches must be augmented with a method for estimation of the model order. Popular choices include generic information theoretic criteria (e.g. AIC, BIC) or more specialized methods, such as SORTE [9] which is based on the eigenvalues of the estimated signal covariance matrix. Subspace methods typically perform extremely well if the model order is known, but their estimation accuracy can degrade significantly if the model order is unknown.

The stochastic maximum likelihood (ML) method is known to be asymptotically efficient (it attains the Cramér-Rao bound as the problem size tends to infinity) [2]. Unfortunately it also requires knowledge of the model order.

Inspired by the ideas of sparse estimation and compressed sensing, many papers on sparsity-based LSE algorithms have appeared in recent years, e.g. [1], [10]. In particular, the LSE problem is simplified to a finite sparse reconstruction problem by restricting the frequencies to a grid. Such methods inherently estimate the model order, alleviating the issues arising from separate model order and frequency estimation in classical methods. The granularity of the grid leads to a non-trivial trade-off between accuracy and computational requirements. To forego the use of a grid, so-called off-the-grid compressed sensing methods have been proposed [11]–[13]. These methods provably recover the frequencies in the noise-free case under a minimum separation condition. They suffer from prohibitively high computational requirements even for moderate problem sizes, see Sec VI.

In [14]–[16] a Bayesian view is taken on the LSE problem. The model used in stochastic ML is extended with a sparsity-promoting prior on the coefficients of the sinusoid components. Thereby inherent estimation of the model order is achieved. These algorithms generally have high estimation accuracy. Their per-iteration computational complexity is cubic in the number of sinusoidal components, meaning that their runtime grows rapidly as the number of components increases.

In this work we introduce the Superfast LSE algorithm for solving the LSE problem in scenarios where the full measurement vector is available (complete data case). The modelling and design of the basic algorithm which we present in Sec. II is based upon the ideas in [14]–[16]. The main novelty resides in the computational aspects of Superfast LSE. The derived method is based upon several techniques: a so-called superfast Toeplitz inversion algorithm [17], [18] (thereof the name of our algorithm), low-complexity Capon beamforming [19], the Gohberg-Semencul formula [20] and non-uniform fast Fourier transforms [21], [22]. The Superfast LSE algorithm has the following virtues: It inherently estimates all model parameters such as the noise variance and model order and it has low per-iteration computational complexity. Specifically it scales as $\mathcal{O}(N \log^2 N)$ where N is the length of the observed vector. We show empirically that it converges after a few iterations (typically less than 20). This means that for large problem sizes our algorithm can have computation time orders of magnitude lower than that of current methods. It does so without any penalty in estimation accuracy. Our numerical experiments show that Superfast LSE has high estimation accuracy across a wide range of scenarios, being on par with or better than state-of-the-art algorithms.

Synergistically and computationally efficiently combining the steps in the algorithm might appear easy after the fact. This is however not the case. Some other LSE algorithms can benefit in terms of computational effort from our approach, yet not to the extent achieved with the proposed algorithm. For instance, the computational methods in Sec. III can be embedded in VALSE [16]. The resulting scheme will have high

T. L. Hansen and B. H. Fleury are with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark. B. D. Rao is with the Electrical and Computer Engineering department, University of California, San Diego.

The work of T. L. Hansen is supported by the Danish Council for Independent Research under grant id DFF–4005–00549.

Digital Object Identifier 10.1109/TSP.2018.2807417

computational complexity due to the variational estimation of the posterior on the frequencies. Note that our algorithm performs on par with VALSE, but at a significantly reduced computational effort.

For completeness we also present a semifast version of the algorithm which works when only a subset of entries in the measurement vector are available. The Semifast LSE algorithm has per-iteration complexity $\mathcal{O}(N\hat{K}^2 + N\log N)$, where $\hat{K}$ is the number of estimated sinusoids. Algorithms with similar per-iteration complexity are derived in [14], [15], [23], [24]. We have observed that our algorithm converges in a smaller number of iterations when compared to the algorithm in [15], thus leading to lower total runtime.

Outline: In Sec. II we present our modelling and algorithm for LSE. Our low-complexity computational methods are presented in Sec. III (complete date case) and IV (incomplete data case). In Sec. V the algorithm is extended to the case of multiple measurement vectors. Numerical experiments are presented in Sec. VI and conclusions are given in Sec. VII.

Notation: We write vectors as $\boldsymbol{a}$ and matrices as $\mathbf{A}$. The ith entry of vector $\boldsymbol{a}$ is denoted a_i or $[\boldsymbol{a}]_i$; the i,jth entry of matrix $\mathbf{A}$ is denoted $\mathbf{A}_{i,j}$. Let $\boldsymbol{b}$ be a binary vector (containing only zeros and ones) of the same dimension as $\boldsymbol{a}$, then $\boldsymbol{a}_{\boldsymbol{b}}$ denotes a vector which contains those entries in $\boldsymbol{a}$ where the corresponding entry in $\boldsymbol{b}$ is one. The Hadamard (entrywise) product is denoted by $\odot$.

II. An Algorithm for Line Spectral Estimation

We now detail the observation model and the specific objective of the LSE problem. The observation vector $\boldsymbol{y} \in \mathbb{C}^M$ contains time-domain samples and is given by

$$\boldsymbol{y} = \sum_{k=1}^{K} \boldsymbol{\Phi}\boldsymbol{\psi}(\tilde{\theta}_k)\tilde{\alpha}_k + \boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}} + \boldsymbol{w}, \tag{1}$$

where the steering vector function $\boldsymbol{\psi}(\theta_k) : [0,1) \to \mathbb{C}^{N\times 1}$ gives a Fourier vector, i.e., it has nth entry $[\boldsymbol{\psi}(\theta_k)]_n \triangleq \exp(j2\pi(n-1)\theta_k)$ for $n = 1,\dots,N$. We also define $\boldsymbol{\Psi}(\boldsymbol{\theta}) \triangleq [\boldsymbol{\psi}(\theta_1),\cdots,\boldsymbol{\psi}(\theta_{\dim(\boldsymbol{\theta})})]$. The measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$ is either the identity matrix ($M = N$, complete data case) or made of a subset of rows of a diagonal matrix ($M < N$, incomplete data case). The vector $\boldsymbol{w}$ is a white Gaussian noise vector with component variance β. The LSE problem is that of recovering the model order K along with the frequency $\tilde{\theta}_k \in [0,1]$ and coefficient $\tilde{\alpha}_k \in \mathbb{C}$ of each component $k = 1,\dots,K$.

A. Estimation Model

The estimation model and inference approach we present in the following are adaptions of ideas currently available in the literature. We have carefully combined these ideas to obtain an iterative scheme which can be implemented with low complexity as described in Secs. III and IV, while achieving a performance comparable to that of state-of-the-art algorithms.

Our algorithm is based on Bayesian inference in an estimation model which approximates (1). Specifically, to enable estimation of the model order K, we follow [14], [16] and employ a model with $K_{\max} \geq K$ components[1]. Each component has an associated activation variable $z_k \in \{0,1\}$ which is set to 0 or 1 to deactivate or activate it. The activation variables are collected in the sparse vector $\boldsymbol{z}$. The effective estimated model order is given by the number of active components. Based on (1) we write our estimation model

$$\boldsymbol{y} = \sum_{k=1}^{K_{\max}} \boldsymbol{\Phi}\boldsymbol{\psi}(\theta_k)z_k\alpha_k + \boldsymbol{w} = \mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}} + \boldsymbol{w}, \tag{2}$$

where $\theta_k \in [0,1)$ and $\alpha_k \in \mathbb{C}$ are frequencies and coefficients for $k = 1,\dots,K_{\max}$ and we have defined $\mathbf{A}(\boldsymbol{\theta}) \triangleq \boldsymbol{\Phi}\boldsymbol{\Psi}(\boldsymbol{\theta})$.

Due to the Gaussian noise assumption we have

$$p(\boldsymbol{y}|\boldsymbol{\alpha},\boldsymbol{z},\boldsymbol{\theta};\beta) = \mathrm{CN}(\boldsymbol{y};\mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}},\beta\mathbf{I}), \tag{3}$$

where $\mathrm{CN}(\boldsymbol{y};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the probability density function of a circularly symmetric complex normal random variable $\boldsymbol{y}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We assume $\beta \in [\varepsilon_\beta,\infty)$, where $\varepsilon_\beta > 0$ is an arbitrarily small constant which guarantees that the likelihood function is bounded below. A Bernoulli prior is used to promote deactivation of some of the components:

$$p(\boldsymbol{z};\zeta) = \prod_{k=1}^{K_{\max}} \zeta^{z_k}(1-\zeta)^{1-z_k}, \tag{4}$$

where $\zeta \in [0,1/2]$ is the activation probability. The restriction $\zeta \leq 1/2$ ensures that the prior is sparsity inducing. The coefficients are assumed to be independent zero-mean Gaussian

$$p(\boldsymbol{\alpha};\boldsymbol{\gamma}) = \prod_{k=1}^{K_{\max}} \mathrm{CN}(\alpha_k;0,\gamma_k), \tag{5}$$

where $\gamma_k \in [0,\infty)$ is the active-component variance. Sparsity-promoting priors have previously been used for both basis selection [25] and LSE [15]. The Bernoulli-Gaussian prior structure that we have adopted above was first introduced in [26] and used for LSE in [16].

Even though each α_k is modelled as Gaussian in (5), the prior specification is significantly more general than that because the variance of each component is estimated through γ_k. In the numerical investigation we demonstrate that our method works well even when the true density of each coefficient is not Gaussian.

We finally use an independent and identically distributed (i.i.d.) uniform prior on the entries in $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}) = \prod_{k=1}^{K_{\max}} p(\theta_k) = \prod_{k=1}^{K_{\max}} 1 = 1. \tag{6}$$

If further prior information about the frequencies is available, it can easily be incorporated through $p(\boldsymbol{\theta})$.

[1]Since we can never expect to estimate more parameters than the number of observed observations, we select $K_{\max} = M$ in our implementation.

B. Approach

By integrating the component coefficients we obtain the marginal likelihood

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\theta};\beta,\boldsymbol{\gamma}) &= \int p(\boldsymbol{y}|\boldsymbol{\alpha},\boldsymbol{z},\boldsymbol{\theta};\beta)p(\boldsymbol{\alpha};\boldsymbol{\gamma})\,\mathrm{d}\boldsymbol{\alpha} \\ &= \mathrm{CN}(\boldsymbol{y};\mathbf{0},\mathbf{C}) \end{aligned} \tag{7}$$

with $\mathbf{C} \triangleq \beta\mathbf{I} + \mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\Gamma}_{\boldsymbol{z}}\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}_{\boldsymbol{z}})$ and $\boldsymbol{\Gamma}_{\boldsymbol{z}} \triangleq \mathrm{diag}(\boldsymbol{\gamma}_{\boldsymbol{z}})$.

Based on the marginal likelihood we can write the objective

$$\begin{aligned} \mathcal{L}(\boldsymbol{z},&\zeta,\beta,\boldsymbol{\theta},\boldsymbol{\gamma}) \triangleq -\ln p(\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{y};\beta,\boldsymbol{\gamma},\zeta) \\ &= -\ln p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\theta};\beta,\boldsymbol{\gamma})p(\boldsymbol{z};\zeta)p(\boldsymbol{\theta}) + \mathrm{const.} \\ &= \ln|\mathbf{C}| + \boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y} \\ &\quad - \sum_{k=1}^{K_{\max}} (z_k \ln\zeta + (1-z_k)\ln(1-\zeta)) + \mathrm{const.} \end{aligned} \tag{8}$$

The variables $(\boldsymbol{z},\boldsymbol{\theta})$ and model parameters $(\beta,\boldsymbol{\gamma},\zeta)$ are estimated by minimizing (8), i.e., we seek the maximum a-posteriori (MAP) estimate of $(\boldsymbol{z},\boldsymbol{\theta})$ and the ML estimate of $(\beta,\boldsymbol{\gamma},\zeta)$. Our algorithm employs a block-coordinate descent method to find a local minimum (or saddle point) of (8).

For fixed $\boldsymbol{z}$ the first two terms in (8) are equal to the objective function of stochastic ML [2], and our approach can therefore be viewed as stochastic ML extended with a variable model order.

When the above estimates have been computed, the estimated model order is given by the number of active components, i.e. $\hat{K} = ||\hat{\boldsymbol{z}}||_0$, and the entries of $\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}$ are the estimated frequencies. The corresponding coefficients $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ can be estimated as follows. First, write the posterior of $\boldsymbol{\alpha}$ as

$$\begin{aligned} p(\boldsymbol{\alpha}|\boldsymbol{y},\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}};\hat{\beta},\hat{\boldsymbol{\gamma}}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha},\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}};\hat{\beta})p(\boldsymbol{\alpha};\hat{\boldsymbol{\gamma}}) \\ &\propto \mathrm{CN}(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}};\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}) \prod_{\{k:\hat{z}_k=0\}} \mathrm{CN}(\alpha_k;0,\hat{\gamma}_k), \end{aligned} \tag{9}$$

where

$$\hat{\boldsymbol{\mu}} \triangleq \hat{\beta}^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\boldsymbol{y} \tag{10}$$

$$\hat{\boldsymbol{\Sigma}} \triangleq \left(\hat{\beta}^{-1}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) + \hat{\boldsymbol{\Gamma}}_{\hat{\boldsymbol{z}}}^{-1}\right)^{-1}. \tag{11}$$

As expected the posterior of the coefficients corresponding to inactive components (those for which $\hat{z}_k = 0$) coincides with their prior. These are not of interest (they are inconsequential in the model (2)) and integrating them out gives a Gaussian posterior over $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$. If a point estimate of $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ is needed, the MAP (which is also the LMMSE) estimate $\hat{\boldsymbol{\alpha}}_{\hat{\boldsymbol{z}}} = \hat{\boldsymbol{\mu}}$ can be used[2].

C. Derivation of Update Equations

As mentioned, our algorithm is derived as a block-coordinate descent method applied on $\mathcal{L}$ in (8). The estimates are updated in the following blocks: $\hat{\boldsymbol{z}}$, $\hat{\zeta}$, $\hat{\beta}$ and $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}},\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}})$. Each update is guaranteed not to increase $\mathcal{L}$. We note that the frequencies and variances of inactive components (those for which $\hat{z}_k = 0$) are not updated, as $\mathcal{L}$ does not depend on these variables.

1) Estimation of frequencies and coefficient variances: Even when all remaining variables are kept fixed, it is not tractable to find the global minimizer of $\mathcal{L}$ with respect to the vector of active component frequencies $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and variances $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$. We therefore resort to a numerical method. Writing only the terms of (8) which depend on $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$, we have

$$\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}) = \ln|\mathbf{C}| + \boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y} + \mathrm{const.},$$

so we need to solve $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}},\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}) = \underset{(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})}{\arg\min}\, \mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$.

In [10] a similar optimization problem involving only the frequencies is solved by Newton's method. Directly applying that approach in our case leads to high computational complexity. Methods based on gradient descent have also been proposed [14], but we have observed that using this approach leads to slow converge. As we are concerned with computational speed in this paper, we instead use the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [27]. This algorithm only requires evaluation of the objective function and its gradient. In the following we demonstrate how these evaluations can be performed with low complexity. At the same time the per-iteration of L-BFGS is linear in $\hat{K}$, namely $\mathcal{O}(J\hat{K})$, where J is the number of saved updates used in L-BFGS. In our implementation we use $J = 10$. We have observed that L-BFGS converges in a small number of iterations.

The L-BFGS algorithm requires an initial estimate of the Hessian of $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$, which is subsequently updated in each iteration of the algorithm. Every update of the activation variable $\hat{\boldsymbol{z}}$ results in a change in the dimension of the Hessian (the number of variables in $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ changes). This means that the implicit estimate of the Hessian in the L-BFGS algorithm is reinitialized rather frequently in our estimation scheme. As a result, the degree of accuracy of the initialization of the Hessian has a significant impact on the convergence speed of the algorithm. We therefore propose to initialize L-BFGS with a diagonal approximation of the Hessian. As shown below, the diagonal entries of the Hessian can be obtained with low computational complexity.

The initial estimate of the Hessian must be positive definite. This is only achieved when all diagonal entries are positive. Those entries of the diagonal Hessian which are negative are therefore replaced with the following values: For entries corresponding to frequency variables we use $(50N)^2$ as the diagonal Hessian and for the entries corresponding to the variance of the kth component we use $[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k^{-2}$. These heuristic values have been determined by considering a diagonally scaled version of the optimization problem (see [28, Sec. 1.3]).

Here follows the required first- and second-order partial derivatives of $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}},\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ evaluated at the current estimates $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}},\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}})$ (see [15] for some hints on how these are obtained):

$$\frac{\partial\mathcal{L}}{\partial[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k} = 2[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k\,\mathrm{Im}\{t_k - q_k^* r_k\} \tag{12}$$

$$\frac{\partial\mathcal{L}}{\partial[\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}]_k} = s_k - |q_k|^2 \tag{13}$$

[2]Note that for computational convenience we write $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}} \odot \boldsymbol{q}$, where $\boldsymbol{q}$ is defined by (16). See the text following (26).

$$\frac{\partial^2 \mathcal{L}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k^2} = 2[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k \mathrm{Re}\big\{ x_k - v_k + [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k (t_k^2 - x_k s_k) + [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k (x_k |q_k|^2 + s_k |r_k|^2 - 2 t_k r_k q_k^*) + (u_k q_k^* - |r_k|^2) \big\} \quad (14)$$

$$\frac{\partial^2 \mathcal{L}}{\partial [\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}]_k^2} = 2 s_k |q_k|^2 - s_k^2, \quad (15)$$

where we have defined vectors

$$\boldsymbol{q} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{y} \quad (16)$$

$$\boldsymbol{r} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{D} \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{y} \quad (17)$$

$$\boldsymbol{s} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{\Phi} \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \Big) \quad (18)$$

$$\boldsymbol{t} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{D} \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{\Phi} \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \Big) \quad (19)$$

$$\boldsymbol{u} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{D}^2 \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{y} \quad (20)$$

$$\boldsymbol{v} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{D}^2 \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{\Phi} \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \Big) \quad (21)$$

$$\boldsymbol{x} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{D} \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}^{-1} \boldsymbol{\Phi} \mathbf{D} \boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \Big). \quad (22)$$

The notation $\mathrm{diag}(\cdot)$ denotes a vector composed of the diagonal entries of the (matrix) argument. The matrix $\hat{\mathbf{C}}$ is that in (7) evaluated at $\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}$, $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$ and $\hat{\beta}$. We have defined the diagonal matrix $\mathbf{D} \triangleq \mathrm{diag}\big([0, 2\pi, 4\pi, \ldots, (N-1)2\pi]^{\mathrm{T}}\big)$. In Secs. III and IV we discuss how the vectors (16)–(22) can be calculated with low computational complexity.

2) Estimation of activation probability: With all other variables fixed, the objective (8) is a convex function of $\zeta \in [0, 1/2]$ $\left(\frac{\partial^2 \mathcal{L}}{\partial \zeta^2} > 0 \right)$. The global minimizer is then found by differentiating and setting equal to zero. Considering the constraints on ζ, we update it as

$$\hat{\zeta} = \min\left(\frac{1}{2}, \frac{||\hat{\boldsymbol{z}}||_0}{K_{\max}} \right). \quad (23)$$

3) Estimation of noise variance: Even when keeping all remaining variables fixed at their current estimate, the globally minimizing noise variance β in (8) cannot be found in closed form. An obvious alternative approach would be to incorporate the estimation of β into L-BFGS together with the estimation of $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$. However, we have observed this approach to exhibit slow convergence because the objective function can be rather "flat" in the variable β (the gradient is small far away from any stationary point).

In sparse Bayesian learning [25] a similar estimation problem is solved successfully via the expectation-minimization (EM) algorithm. To use EM, we need to reintroduce $\boldsymbol{\alpha}$ into the estimation problem. In order to show how EM is integrated into our coordinate-block descent method and that the update of $\hat{\beta}$ is guaranteed not to increase (8), it is the easiest to directly use the upper bound associated with EM (see [29] for a derivation of EM which takes a similar approach).

The updated estimate of β is the minimizer of an upper bound on the objective function (8). To obtain the upper bound we write the terms of the objective function which depend on β, with all other variables kept fixed at their current estimates:

$$\begin{aligned} \mathcal{L}(\beta) &= -\ln p(\boldsymbol{y} | \hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}}) + \mathrm{const.} \\ &= -\ln \int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \frac{p(\boldsymbol{y}, \boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} | \hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}})}{f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})} \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} + \mathrm{const.} \\ &\leq -\int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} | \hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}})}{f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})} \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} + \mathrm{const.}, \end{aligned} \quad (24)$$

where $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \geq 0$ is a function which fulfills $\int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} = 1$. The inequality follows from Jensen's inequality.

Following EM, we select $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) = p(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} | \boldsymbol{y}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \hat{\beta}^{i-1}, \hat{\boldsymbol{\gamma}})$, where $\hat{\beta}^{i-1}$ denotes the previous noise variance estimate. Denote the upper bound on the right-hand side of (24) by $Q(\beta; \hat{\beta}^{i-1})$ and insert $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})$ to get

$$Q(\beta; \hat{\beta}^{i-1}) = M \ln \beta + \beta^{-1} \, \mathrm{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \right) + \beta^{-1} ||\boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \hat{\boldsymbol{\mu}}||^2 + \mathrm{const.}, \quad (25)$$

where we have used (9) to evaluate expectations involving $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ and $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are calculated from (10)–(11) based on $\hat{\beta}^{i-1}$. It is easy to show that the upper bound has a unique minimizer, which is used as the updated estimate of the noise variance:

$$\hat{\beta}^i = \max\left(\varepsilon_\beta, \frac{\mathrm{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \right) + ||\boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \hat{\boldsymbol{\mu}}||^2}{M} \right). \quad (26)$$

To allow low-complexity calculation of $\hat{\beta}^i$ we use Woodbury's matrix inversion identity to show that $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}} \odot \boldsymbol{q}$ and $\mathrm{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \right) = \sum_{k=1}^{\hat{K}} (\hat{\beta}^{i-1} s_k [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k)$.

The update (26) could be applied repeatedly since an improved upper bound is used each time. Since we have not observed any advantages by doing so, we simply perform the update (26) once for each pass in the block-coordinate descent algorithm. We also note that even though EM is known to be prone to slow convergence speed, we have observed empirically that the estimate of β converges fast, typically within 10 iterations.

It can easily be shown that with the chosen $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})$, the inequality in (24) holds with equality at $\beta = \hat{\beta}^{i-1}$. It then follows that the new estimate of β does not increase the value of the objective function (see the proof of Lemma 4 in Appendix A).

4) Deactivation of Components: We now describe the activation and deactivation of components, which is performed by the single most likely replacement (SMLR) detector [26]. SMLR has previously been demonstrated to perform well for LSE [14]–[16], [23].

First we write the terms of (8) which depend on the variables pertaining to the kth component and fix all other variables at their current estimate. Based on Woodbury's matrix inversion identity and the determinant lemma we get (see [24] for details)

$$\mathcal{L}(z_k, \theta_k, \gamma_k) = z_k \left(-\frac{|q_{\sim k}(\theta_k)|^2}{\gamma_k^{-1} + s_{\sim k}(\theta_k)} + \ln\left((1 + \gamma_k s_{\sim k}(\theta_k)) \frac{1 - \hat{\zeta}}{\hat{\zeta}} \right) \right) + \mathrm{const.}, \quad (27)$$

with

$$q_{\sim k}(\theta_k) \triangleq \boldsymbol{\psi}^{\mathrm{H}}(\theta_k)\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}_{\sim k}^{-1}\boldsymbol{y} \qquad (28)$$
$$s_{\sim k}(\theta_k) \triangleq \boldsymbol{\psi}^{\mathrm{H}}(\theta_k)\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}_{\sim k}^{-1}\boldsymbol{\Phi}\boldsymbol{\psi}(\theta_k),$$

where $\hat{\mathbf{C}}_{\sim k} \triangleq \hat{\beta}\mathbf{I}+\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}_{\sim k}})\hat{\boldsymbol{\Gamma}}_{\hat{\boldsymbol{z}}_{\sim k}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}_{\sim k}})$ and $\hat{\boldsymbol{z}}_{\sim k}$ is equal to $\hat{\boldsymbol{z}}$ with the kth entry forced to zero. The matrix $\hat{\mathbf{C}}_{\sim k}$ is thus the marginal covariance matrix of the observation vector with the kth component deactivated.

To evaluate if an active component should be deactivated, we test if the objective $\mathcal{L}$ is increased by doing so, i.e., we test if $\mathcal{L}(z_k = 0, \hat{\theta}_k, \hat{\gamma}_k) < \mathcal{L}(z_k = 1, \hat{\theta}_k, \hat{\gamma}_k)$. This gives the deactivation criterion for the kth component:

$$\frac{|q_{\sim k}(\hat{\theta}_k)|^2}{\hat{\gamma}_k^{-1} + s_{\sim k}(\hat{\theta}_k)} - \ln\Big(1 + \hat{\gamma}_k s_{\sim k}(\hat{\theta}_k)\Big) < \ln\left(\frac{1-\hat{\zeta}}{\hat{\zeta}}\right). \quad (29)$$

This criterion is evaluated for currently active components, i.e., for k which has corresponding $\hat{z}_k = 1$.

For computational convenience we note that we can obtain $q_{\sim k}(\hat{\theta}_k)$ and $s_{\sim k}(\hat{\theta}_k)$ from $\boldsymbol{q}$ and $\boldsymbol{s}$ with low complexity. First, write $\hat{\mathbf{C}}_{\sim k} = \hat{\mathbf{C}} - \hat{\gamma}_k\boldsymbol{\Phi}\boldsymbol{\psi}(\hat{\theta}_k)\boldsymbol{\psi}^{\mathrm{H}}(\hat{\theta}_k)\boldsymbol{\Phi}^{\mathrm{H}}$ and use Woodbury's identity to obtain

$$q_{\sim k}(\hat{\theta}_k) = \frac{q_i}{1-\hat{\gamma}_k s_i}$$
$$s_{\sim k}(\hat{\theta}_k) = \frac{s_i}{1-\hat{\gamma}_k s_i},$$

where q_i and s_i are the ith entries of (16) and (18) with i denoting the index for which $[\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}]_i = \hat{\theta}_k$.

5) Component Activation: We now describe a method to decide if a deactivated component should be activated. This also involves estimating the frequency and variance of this component, because no meaningful such estimates are available before the component is activated. Any of the deactivated components are equally good candidates for activation. In the following k refers to an arbitrary value for which $\hat{z}_k = 0$. If no such k exists all components are already activated and the activation step is not carried out.

Our method is again based on the expression (27). Inspired by [16], let $\bar{\gamma}$ denote the average of the entries in $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$. Define the change in the objective obtained from setting $\hat{z}_k = 1$, $\hat{\theta}_k = \theta_k$, $\hat{\gamma}_k = \bar{\gamma}$:

$$\Delta\mathcal{L}(\theta_k) = \mathcal{L}(1, \theta_k, \bar{\gamma}) - \mathcal{L}(0, \theta_k, \bar{\gamma}) \qquad (30)$$
$$= \ln\left((1+\bar{\gamma}s_{\sim k}(\theta_k))\,\frac{1-\hat{\zeta}}{\hat{\zeta}}\right) - \frac{|q_{\sim k}(\theta_k)|^2}{\bar{\gamma}^{-1} + s_{\sim k}(\theta_k)}$$

Note that the last term in (30) does not depend on θ_k or $\bar{\gamma}$. Then the frequency is found by maximizing the decrease in the objective, i.e.,

$$\hat{\theta}_k = \underset{\theta_k \in \mathcal{G}}{\arg\min}\ \Delta\mathcal{L}(\theta_k), \qquad (31)$$

where $\mathcal{G}$ is a grid of L equispaced values, i.e., $\mathcal{G} \triangleq \{0, 1/L, \ldots, 1-1/L\}$. The restriction of the estimated frequencies to a grid does not mean that the final frequency estimates lie on a grid, because they are refined to be in $[0, 1)$ in subsequent updates of the frequency vector. For this reason, the choice of L does not have any impact on the estimation accuracy, provided that it is sufficiently large.[3] In Sec. III and IV we show how $q_{\sim k}(\theta_k)$ and $s_{\sim k}(\theta_k)$ can be evaluated with low complexity for all $\theta_k \in \mathcal{G}$, such that the minimization can be performed by means of an exhaustive search over $\mathcal{G}$.

The activation procedure continues only if a decrease in the objective can be obtained by activating a component at $\hat{\theta}_k$, i.e., if $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$. The inclusion of the constant $\varepsilon_{\mathcal{L}} > 0$ is purely technical, as it simplifies our convergence analysis. It can be chosen arbitrarily small and we select it as machine precision in our implementation.

After estimating the frequency, the component variance is selected as $\hat{\gamma}_k = \underset{\gamma_k}{\arg\min}\ \Delta\mathcal{L}(1, \hat{\theta}_k, \gamma_k)$. Using an approach similar to [24], this minimizer can be shown to be

$$\hat{\gamma}_k = \begin{cases} \frac{|q_{\sim k}(\theta_k)|^2 - s_{\sim k}(\theta_k)}{s_{\sim k}^2(\theta_k)} & \text{if } \frac{|q_{\sim k}(\theta_k)|^2}{s_{\sim k}(\theta_k)} > 1, \\ 0 & \text{otherwise.} \end{cases} \qquad (32)$$

The component is only activated if[4] $\hat{\gamma}_k > 0$.

It is instructive to explore the activation criterion $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$ in detail. Since $\varepsilon_{\mathcal{L}}$ is machine precision, we ignore it ($\varepsilon_{\mathcal{L}} = 0$) for simplicity. The activation criterion can be rewritten to the form

$$\frac{|q_{\sim k}(\hat{\theta}_k)|^2}{s_{\sim k}(\hat{\theta}_k)} > \left(1 + \frac{1}{\bar{\gamma}s_{\sim k}(\hat{\theta}_k)}\right)\ln\left(\Big(1+\bar{\gamma}s_{\sim k}(\hat{\theta}_k)\Big)\,\frac{1-\hat{\zeta}}{\hat{\zeta}}\right). \qquad (33)$$

Denote the left-hand side of (33) as κ_k. This quantity can be interpreted as the signal-to-noise ratio of the kth component [30], [31]. If the sparse Bayesian learning (SBL) model is used for sparsity promotion, an activation criterion of the from $\kappa_k > 1$ is obtained [30], [31]. Algorithms using the activation criterion $\kappa_k > 1$ are known to be prone to the activation of "artefact" components with very small $\hat{\gamma}_k$ and $\hat{\alpha}_k$ at what seems to be arbitrary frequencies $\hat{\theta}_k$. The right-hand side of (33) is always larger than one and this helps reduce the number of artefacts which are activated, as demonstrated in [16]. This favorable phenomenon is caused by the use of the average $\bar{\gamma}$ in the definition of $\Delta\mathcal{L}(\theta_k)$ (as opposed to inserting $\hat{\gamma}_k$ from (32), which resembles the SBL approach).

Even still, we have observed the activation of a few artefact components in our numerical investigations. We therefore follow the same idea as [30], [31] and heuristically adjust the criterion (33) to obtain

$$\frac{|q_{\sim k}(\theta_k)|^2}{s_{\sim k}(\theta_k)} > \left(1 + \frac{1}{\bar{\gamma}s_{\sim k}(\theta_k)}\right)\ln\left((1+\bar{\gamma}s_{\sim k}(\theta_k))\,\frac{1-\hat{\zeta}}{\hat{\zeta}}\right) + \tau, \qquad (34)$$

where $\tau \geq 0$ is some adjustment of the threshold. Specifically we select $\tau = 5$, cf. the numerical study in Sec. VI-B. Our numerical experiments show that this simple approach is very effective at avoiding the inclusion of small spurious components. Since the heuristic criterion (34) is stricter than the

[3] A numerical investigation (not reported here) shows that the algorithm is invariant to the choice of L, provided that $L \geq 2N$. In our implementation we use L equal to $8N$ rounded to the nearest power of 2.

[4] When $\hat{\gamma}_k = 0$ the kth component is effectively deactivated because the corresponding coefficient α_k has a zero-mean prior with zero variance, see (5). The effective deactivation is also seen in the definition of $\mathbf{C}$ in (7) and it further manifests itself as $\hat{\mu}_k = 0$ in (10).

criterion $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$, it is guaranteed that the activation of a component decreases the objective function.

D. Outline of the Algorithm and Implementation Details

The algorithm proceeds by repeating the following steps until convergence:

1) Check if any components can be activated via the procedure described in Sec. II-C5.
2) Re-estimate the activation probability ζ via (23).
3) Re-estimate the noise variance via (26).
4) Repeat:
   4a) Perform a single L-BFGS update of the estimated vectors of active component frequencies $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and variances $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$, as described in Sec. II-C1.
   4b) Check if any components can be deactivated via (29).

The algorithm terminates when the change in the objective (8) between two consecutive iterations is less than $M10^{-7}$.

In step 1) and 4b) the check for component (de)activation is repeated until no more components can be (de)activated. The updates in step 4) are iterated until either the approximated squared Newton decrement of the L-BFGS method is below $M10^{-8}$ or at most 5 times.

The observant reader will have noticed that the minimization over $(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ must be constrained to $\gamma_k \geq 0$ for all k. It turns out that this constraint can be handled in a simple manner: Notice that the deactivation criterion (29) is always fulfilled for $\hat{\gamma}_k$ sufficiently small. The constraint is therefore never active at the solution. We therefore simply need to restrict the line-search performed in L-BFGS such that the no entry in $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$ ever becomes negative. If any $\hat{\gamma}_k$ approaches (or becomes equal to) zero, it is deactivated in step 4b). Note that this approach resembles that of L-BFGS for box constraints [32], except that the deactivation of variables for which the constraint is active happens automatically in our algorithm.

The algorithm is initialized with all components in the deactivated stage (i.e. $\hat{\boldsymbol{z}} = \boldsymbol{0}$). The initial values of the entries in $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\gamma}}$ do not matter, since they are assigned when their corresponding component is activated (see Sec. II-C5). The noise variance is initialized to $\hat{\beta} = 0.01||\boldsymbol{y}||^2/M$ ($1\,\%$ of the energy in $\boldsymbol{y}$ is assumed to be noise). The activation probability is initialized to $\hat{\zeta} = 0.2$.

In Appendix A we discuss in detail the convergence properties of our algorithm. The findings are summarized here. We show that our algorithm terminates in finite time and that the estimates of $\boldsymbol{z}$, ζ and β are guaranteed to converge. We denote the limit points as $\bar{\boldsymbol{z}}$, $\bar{\zeta}$ and $\bar{\beta}$. When these estimates have converged, our algorithm reduces to a pure L-BFGS scheme which estimates $(\boldsymbol{\theta}_{\bar{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\bar{\boldsymbol{z}}})$. Due to the non-convexity of the objective function, we cannot guarantee convergence of L-BFGS (see [33]). Despite of this, we have never observed non-convergence of our algorithm. In our experiments it always converged to a local minimum of the objective function. We therefore rely on the vast amount of experimental validation of the convergence of L-BFGS and assume convergence to a stationary point. In particular, we have the following theorem.

Theorem 1: Assume that L-BFGS in step 4a) converges to a stationary point of $(\boldsymbol{\theta}_{\bar{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\bar{\boldsymbol{z}}}) \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \bar{\zeta}, \bar{\beta}, \boldsymbol{\theta}_{\bar{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\bar{\boldsymbol{z}}})$. Then the sequence of estimates obtained by our algorithm converges. Further, the limit point is a stationary point of $(\zeta, \beta, \boldsymbol{\theta}, \boldsymbol{\gamma}) \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \zeta, \beta, \boldsymbol{\theta}, \boldsymbol{\gamma})$, in the sense that the Karush-Kuhn-Tucker necessary conditions for a minimum are fulfilled.

Proof: See Appendix A. ■

E. Initial Activation of Components

When the number of sinusoids K in the observed signal (1) is high, the algorithm spends significant computational effort activating components (step 1). This is because each time a component is activated, the values $q_{\sim k}(\theta_k)$ and $s_{\sim k}(\theta_k)$ must be evaluated for all $\theta_k \in \mathcal{G}$ to calculate (31). To alleviate the computational effort of building the initial set of active components, we propose to let the first few iterations use an approximate scheme for activating components in place of step 1). The approximate activation scheme proceeds as follows:

1) Calculate $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$, where k is the index of a deactivated component.
2) Evaluate $\Delta\mathcal{L}(\theta)$ (30) for all $\theta \in \mathcal{G}$.
3) Find the local minimizers of $\Delta\mathcal{L}(\theta)$, i.e., find the values of θ for which $\Delta\mathcal{L}(\theta) \leq \Delta\mathcal{L}(\theta')$ with θ' being any of the two neighbouring grid-points of θ. The local minimizers are candidate frequencies.
4) Activate a component at those candidate frequencies for which the following criteria are fulfilled:
   - The component activation criterion (33) is fulfilled.
   - The component variance (32) is non-zero.
   - The decrease in the objective obeys $\Delta\mathcal{L}(\theta) \leq \Delta\mathcal{L}_{\text{min}}/5$, where $\Delta\mathcal{L}_{\text{min}}$ is the largest decrease obtained from activating a component at another candidate frequency (in the current iteration).
   - All other currently active components have frequency estimates located at least[5] $0.05N^{-1}$ apart from the candidate frequency.

The above method is a heuristic scheme, which quickly builds a set of activated components. Typically this set is close to the final result and only a few (in our setup less than 15 in most cases) iterations are need before convergence.

III. Superfast Method (Complete Observations)

The algorithm presented above has rather large computational complexity, in particular due to the inversion of $\mathbf{C}$ and the calculation of $q_{\sim k}(\theta)$ $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$. In this section we discuss how all updates of the algorithm can be evaluated with low computational complexity by exploiting the inherent structure of the problem. In particular we discuss how to evaluate $\ln|\mathbf{C}|$, $\boldsymbol{y}^{\text{H}}\mathbf{C}^{-1}\boldsymbol{y}$, $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{x}$ and $q_{\sim k}(\theta)$, $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$.

The method presented here is only applicable when the complete observation vector is available, i.e., when $\boldsymbol{\Phi} = \mathbf{I}$, $M = N$ and $\mathbf{A}(\boldsymbol{\theta}) = \boldsymbol{\Psi}(\boldsymbol{\theta})$. In this case the observation vector $\boldsymbol{y}$ is a wide-sense stationary process and its covariance

[5]For the distance measure we use the wrap-around distance on $[0, 1)$ defined as $d(x, y) \triangleq \min(|x - y|, 1 - |x - y|)$ for $x, y \in [0, 1)$.

matrix $\mathbf{C}$ is Hermitian Toeplitz. Low-complexity algorithms for inverting such matrices are available in the literature. We also rely on fast Fourier transform (FFT) techniques.

Our approach is based on the Gohberg-Semencul formula [17], [20], which states that the inverse of the Hermitian Toeplitz matrix $\mathbf{C}$ can be decomposed as

$$\mathbf{C}^{-1} = \delta_{N-1}^{-1}\left(\mathbf{T}_1^{\mathrm{H}}\mathbf{T}_1 - \mathbf{T}_0\mathbf{T}_0^{\mathrm{H}}\right), \tag{35}$$

where the entries of $\mathbf{T}_0$ and $\mathbf{T}_1$ are

$$[\mathbf{T}_0]_{i,k} = \rho_{i-k-1},$$
$$[\mathbf{T}_1]_{i,k} = \rho_{N-1+i-k}$$

for $i,k = 1,\ldots,N$. Note that $\rho_i = 0$ for $i < 0$ and $i > N-1$; thus $\mathbf{T}_0$ is strictly lower triangular and $\mathbf{T}_1$ is unit upper triangular ($\rho_{N-1} = 1$). The values δ_i and ρ_i for $i = 0,\ldots,N-1$ can be computed with a generalized Schur algorithm in time $\mathcal{O}(N\log^2 N)$ [17]. Alternatively, the Levinson-Durbin algorithm can also be used to obtain the decomposition in time $\mathcal{O}(N^2)$. The latter algorithm is significantly simpler to implement and is faster for small N. In [18] it is concluded that the Levinson-Durbin algorithm requires fewer total operations than the generalized Schur algorithm for $N \leq 256$.

A. Evaluating $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ and $\ln|\mathbf{C}|$

To calculate the value of the objective function (8) we need to find $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ and $\ln|\mathbf{C}|$. Inspecting (35) it is clear that matrix-vector products involving $\mathbf{T}_0$ and $\mathbf{T}_1$ are convolutions. These can be implemented using FFT techniques. The product $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ can thus be calculated in $\mathcal{O}(N\log N)$ time when $\{\rho_i\}$ and δ_{N-1} are known.

The matrix $\mathbf{C}$ is Hermitian positive definite and can therefore be factorized uniquely as

$$\mathbf{C} = \mathbf{L}\mathbf{B}\mathbf{L}^{\mathrm{H}}, \tag{36}$$

with $\mathbf{L}$ being unit lower triangular. The diagonal matrix $\mathbf{B}$ is computed with the generalized Schur algorithm. Its diagonal entries are given by δ_i for $i = 0,\ldots,N-1$ [17]. Since the determinant of a triangular matrix is the product of its diagonal entries, we have

$$\ln|\mathbf{C}| = \sum_{i=0}^{N-1}\ln\delta_i. \tag{37}$$

It follows that once the generalized Schur algorithm has been executed, the objective function (8) can easily be found.

B. Evaluating $\boldsymbol{q}$, $\boldsymbol{r}$ and $\boldsymbol{u}$

Note that $\mathbf{C}^{-1}\boldsymbol{y}$ can be evaluated with FFT techniques using (35). We recognize that matrix-vector products involving $\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ are Fourier transforms evaluated off the equispaced grid. Such products are approximated to a very high precision in time $\mathcal{O}(N\log N)$ using the non-uniform fast Fourier transform[6] (NUFFT) [21], [22]. Then $\boldsymbol{q}$, $\boldsymbol{r}$ and $\boldsymbol{u}$ are easily found in time $\mathcal{O}(N\log N)$ (assuming the decomposition (35) has already been calculated).

C. Evaluating $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{v}$ and $\boldsymbol{x}$

Turning our attention to $\boldsymbol{s}$, we follow [19] and note that (recall that we assume $\boldsymbol{\Phi} = \mathbf{I}$)

$$\begin{aligned} s_k &= [\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{C}^{-1}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})]_{k,k} \\ &= \sum_{i=-(N-1)}^{N-1}\omega_s(i)\exp\Big(j2\pi i[\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}]_k\Big) \end{aligned} \tag{38}$$

for $k = 1,\ldots,\hat{K}$ where $\hat{K}$ is the number of entries in $\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}$. The function $\omega_s(i)$ gives the sum over the ith diagonal, i.e.,

$$\omega_s(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)}[\mathbf{C}^{-1}]_{q+1,q+i+1}. \tag{39}$$

It is obvious that (38) can be calculated for all $k = 1,\ldots,\hat{K}$ via a NUFFT when the values $\omega_s(i)$ are available.

To evaluate $\boldsymbol{t}$, $\boldsymbol{v}$ and $\boldsymbol{x}$ we follow a similar approach and note that the entries of these vectors can be written as (38) with $\omega_s(i)$ replaced by

$$\omega_t(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)}[\mathbf{D}\mathbf{C}^{-1}]_{q+1,q+i+1} \tag{40}$$

$$\omega_v(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)}[\mathbf{D}^2\mathbf{C}^{-1}]_{q+1,q+i+1} \tag{41}$$

$$\omega_x(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)}[\mathbf{D}\mathbf{C}^{-1}\mathbf{D}]_{q+1,q+i+1}, \tag{42}$$

respectively. In Appendix B we demonstrate how $\{\omega_s(i)\}$, $\{\omega_t(i)\}$, $\{\omega_v(i)\}$ and $\{\omega_x(i)\}$ can be obtained through length-$2N$ FFTs using the decomposition (35).

D. Evaluating $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$

To calculate the frequency of the component processed in the activation stage, $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ must be evaluated for all $\theta \in \mathcal{G}$, where $\mathcal{G}$ is a grid of L equispaced points. Defining the vector of gridded frequencies $\boldsymbol{\theta}^{\mathcal{G}} \triangleq [0, 1/L, \ldots, (L-1)/L]^{\mathrm{T}}$, we need to find

$$\begin{aligned} \boldsymbol{q}^{\mathcal{G}} &\triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{C}^{-1}\boldsymbol{y}, \\ \boldsymbol{s}^{\mathcal{G}} &\triangleq \mathrm{diag}\left(\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{C}^{-1}\boldsymbol{\Psi}(\boldsymbol{\theta}^{\mathcal{G}})\right). \end{aligned}$$

We have used the fact that in the beginning of the activation step the kth component is deactivated and thus $\mathbf{C}_{\sim k} = \mathbf{C}$.

[6]The NUFFT calculates the Fourier transform at arbitrary points (not lying on an equispaced grid) by interpolation combined with an FFT. It is an approximation, which can be made arbitrarily accurate by including more points in the interpolation. The NUFFT achieves a time complexity of $\mathcal{O}(N\log N + K)$, where K is the number of off-the-grid frequency points at which it is evaluated. For $K \leq N$ this complexity is equal to that of the FFT, but the constant hidden in the big-O notation is much higher for the NUFFT. We have found that for $N \geq 512$ significant speedups can be achieved by using the NUFFT over a direct computation of $\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ and evaluation of the matrix-vector products involving this matrix. In particular the speedup arises from the fact that $\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ no longer needs to be formed.

Since $\mathcal{G}$ is an equispaced grid, products with $\mathbf{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})$ can be evaluated as a length-L FFT. The vector $\boldsymbol{q}^{\mathcal{G}}$ is therefore easy to find. Rewriting $\boldsymbol{s}^{\mathcal{G}}$ in the form (38), it is seen that $\boldsymbol{s}^{\mathcal{G}}$ can also be evaluated as a length-L FFT. These computations have time-complexity $\mathcal{O}(L \log L)$ (assuming the decomposition (35) has already been calculated).

E. Algorithm Complexity

In summary, the time complexity of each iteration in the algorithm described in Sec. II is dominated by either the calculation of $\{\rho_i\}$ and δ_{N-1} with the generalized Schur algorithm or the calculation of $\boldsymbol{q}^{\mathcal{G}}$ and $\boldsymbol{s}^{\mathcal{G}}$ (we assume $\hat{K} \leq M = N \leq L$). With our choice $L = 8N$ we have complexity per iteration of $\mathcal{O}(N \log^2 N)$.

Also note that all computations involving $\mathbf{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ are performed using the NUFFT. This matrix therefore does not need to be stored, so our algorithm only uses a modest amount of memory.

IV. Semifast Method (Incomplete Observations)

The method presented in Sec. III is not applicable when an incomplete observation vector is available, i.e., when $\mathbf{\Phi} \neq \mathbf{I}$. In the following we introduce a computational method, which can be used when $\mathbf{\Phi}$ is a subsampling and scaling matrix, i.e., when $\mathbf{\Phi} \in \mathbb{C}^{M \times N}$ consists of M rows of a diagonal matrix.[7] With this method we can still obtain an algorithm with reasonable computational complexity per iteration, assuming that $\hat{K}$ is relatively small (a $\hat{K} \times \hat{K}$ matrix must be inverted). We coin this algorithm as semifast. For small $\hat{K}$ the semifast algorithm is faster than the superfast algorithm of Sec. III and it may therefore be beneficial to even use it in the complete data case.

The semifast method is based on the following decomposition of $\mathbf{C}^{-1}$, obtained using Woodbury's matrix identity:

$$\mathbf{C}^{-1} = \hat{\beta}^{-1}\mathbf{I} - \hat{\beta}^{-2}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\hat{\mathbf{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \tag{43}$$

with $\hat{\mathbf{\Sigma}}$ given by (11). We can evaluate $\hat{\mathbf{\Sigma}}^{-1}$ by noting that

$$\begin{aligned}\Big[\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big]_{i,k} &= \Big[\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{\Phi}^{\mathrm{H}}\mathbf{\Phi}\mathbf{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big]_{i,k} \\ &= \sum_{m=1}^{M} |\mathbf{\Phi}_{m,I_{\mathcal{M}}(m)}|^2 \exp\Big(j2\pi(I_{\mathcal{M}}(m)-1)(\hat{\theta}_k - \hat{\theta}_i)\Big),\end{aligned} \tag{44}$$

which can be evaluated with a NUFFT in time $\mathcal{O}(N \log N + \hat{K}^2)$. Forming $\hat{\mathbf{\Sigma}}^{-1}$ is then easy and an inversion[8] in time $\mathcal{O}(\hat{K}^3)$ is needed to obtain $\hat{\mathbf{\Sigma}}$. The approach thus hinges on $\hat{K}$ being sufficiently small, such that the inverse (really, the Cholesky decomposition) can be calculated in reasonable time.

[7] Let $\mathcal{M} \subseteq \{1, \ldots, N\}$ denote the index set of the observed entries and $I_{\mathcal{M}} : \{1, \ldots, M\} \to \mathcal{M}$ be an indexing. Then $\mathbf{\Phi}_{m,I_{\mathcal{M}}(m)}$, $m = 1, \ldots, M$, are the only nonzero elements of $\mathbf{\Phi}$.

[8] As is customary in numerical linear algebra, we would recommend not to explicitly evaluate the inverse, but instead use the numerically stabler and faster approach of calculating the Cholesky decomposition $\hat{\mathbf{\Sigma}}^{-1} = \mathbf{L}\mathbf{L}^{\mathrm{H}}$ (a unique Cholesky decomposition exists because $\hat{\mathbf{\Sigma}}^{-1}$ is Hermitian positive definite). We need to evaluate matrix-vector products involving $\hat{\mathbf{\Sigma}}$ which are easily evaluated from the decomposition by forward-backward substitution. We can also calculate $|\hat{\mathbf{\Sigma}}^{-1}|$ directly from the Cholesky decomposition.

A. Evaluating $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$, $\ln|\mathbf{C}|$, $\boldsymbol{q}$, $\boldsymbol{r}$ and $\boldsymbol{u}$

Notice that matrix-vector products involving $\mathbf{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ and $\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ can be evaluated using a NUFFT. It then immediately follows that the values $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$, $\boldsymbol{q}$, $\boldsymbol{r}$ and $\boldsymbol{u}$ can be evaluated using (43) with complexity $\mathcal{O}(\hat{K}^2 + N \log N)$.

To evaluate the objective function (8) we need to calculate $\ln|\mathbf{C}|$. By invoking the matrix determinant lemma we get

$$\ln|\mathbf{C}| = M \ln \hat{\beta} + \sum_{\{k:\hat{z}_k=1\}} \ln \hat{\gamma}_k + \ln|\mathbf{\Sigma}^{-1}|, \tag{45}$$

which can be evaluated in time $\mathcal{O}(\hat{K})$ once the Cholesky decomposition of $\mathbf{\Sigma}^{-1}$ is known.

B. Evaluating $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{v}$ and $\boldsymbol{x}$

As an example, we demonstrate how to evaluate $\boldsymbol{t}$. We note that $\boldsymbol{s}$, $\boldsymbol{v}$ and $\boldsymbol{x}$ can easily be obtained using the same approach. First, insert (43) into (19) to get

$$\begin{aligned}\boldsymbol{t} = {} & \hat{\beta}^{-1} \operatorname{diag}\Big(\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big) \\ & - \hat{\beta}^{-2} \operatorname{diag}\Big(\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\hat{\mathbf{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big).\end{aligned}$$

Using the same methodology as for computing $\hat{\mathbf{\Sigma}}^{-1}$, the $\hat{K} \times \hat{K}$ matrices $\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ and $\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})$ can be obtained in time $\mathcal{O}(N \log N + \hat{K}^2)$. Then, $\boldsymbol{t}$ is found by direct evaluation in time $\mathcal{O}(\hat{K}^3)$.

C. Evaluating $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$

To calculate the frequency of the component processed in the activation stage we must evaluate $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$, where $\mathcal{G}$ is a grid of L equispaced points. Using the fact that in the beginning of the activation step the kth component is deactivated and thus $\mathbf{C}_{\sim k} = \mathbf{C}$, we obtain the required quantities by inserting (43) into (16) and (18):

$$\begin{aligned}\boldsymbol{q}^{\mathcal{G}} &= \hat{\beta}^{-1}\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\Big(\boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\hat{\boldsymbol{\mu}}\Big) \\ \boldsymbol{s}^{\mathcal{G}} &= \hat{\beta}^{-1} \operatorname{diag}\big(\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})\big) \\ &\quad - \hat{\beta}^{-2} \operatorname{diag}\Big(\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\hat{\mathbf{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})\Big).\end{aligned}$$

It is clear that $\boldsymbol{q}^{\mathcal{G}}$ can easily be found using FFT techniques.

To obtain $\boldsymbol{s}^{\mathcal{G}}$ we first note that the first term is a vector with all entries equal to $\hat{\beta}^{-1}\sum_{m=1}^{M}|\mathbf{\Phi}_{m,I_{\mathcal{M}}(m)}|^2$. The second term is found by using a NUFFT (see (44)) to form $\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})$. Then by using the Cholesky decomposition of $\hat{\mathbf{\Sigma}}^{-1}$ the second term can be calculated in time $\mathcal{O}(L\hat{K}^2)$.

D. Algorithm Complexity

The above computation is dominated by either the calculation of $\boldsymbol{s}^{\mathcal{G}}$ or the length-L FFT involved in calculating $\boldsymbol{q}^{\mathcal{G}}$. Again with $L = 8N$ we have overall complexity per iteration $\mathcal{O}(N\hat{K}^2 + N \log N)$.

V. Multiple Measurement Vectors

The algorithm presented in Sec. II assumes a single measurement vector (SMV). We now discuss an extension to the case of multiple measurement vectors (MMV) [34]. This case is of particular importance in array processing where the number of observation points M is determined by the number of antennas in the array.[9] Typically M is small, which thus limits estimation accuracy. On the other hand it is often easy to obtain multiple observation vectors across which the entries in $\tilde{\boldsymbol{\theta}}$ (the true directions of arrivals) are practically unchanged. The MMV signal model reads

$$\boldsymbol{y}^{(g)} = \mathbf{A}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}^{(g)} + \boldsymbol{w}^{(g)}, \tag{46}$$

where $g = 1, \ldots, G$ indexes the observation vectors.

To extend our SMV algorithm to the MMV case we again impose an estimation model of the form (2) that contains $K_{\max}$ components which can be (de)activated based on variables z_k, $k = 1, \ldots, K_{\max}$. The likelihood for each of the G observation vectors then reads

$$p(\boldsymbol{y}^{(g)}|\boldsymbol{\alpha}^{(g)}, \boldsymbol{z}, \boldsymbol{\theta}; \beta) = \mathrm{CN}(\boldsymbol{y}^{(g)}; \mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}}^{(g)}, \beta\mathbf{I}). \tag{47}$$

We impose the same prior as used in the SMV case (5) on each $\boldsymbol{\alpha}^{(g)}$:

$$p(\boldsymbol{\alpha}^{(g)}; \boldsymbol{\gamma}) = \prod_{k=1}^{K_{\max}} \mathrm{CN}(\alpha_k^{(g)}; 0, \gamma_k). \tag{48}$$

The vectors $\boldsymbol{z}$ and $\boldsymbol{\theta}$ are assigned the same priors as in the SMV case, i.e., as given by (4) and (6). Similarly to the SMV case, the MMV model has parameters $\boldsymbol{\gamma}$, β and ζ.

The objective to be minimized is the marginal likelihood, which for the MMV model reads

$$\begin{aligned}\mathcal{L}_{\text{MMV}} &\triangleq -\ln \prod_{g=1}^{G} p(\boldsymbol{y}^{(g)}|\boldsymbol{z}, \boldsymbol{\theta}; \beta, \boldsymbol{\gamma}) p(\boldsymbol{z}; \zeta) p(\boldsymbol{\theta}) + \text{const.}, \\ &= \sum_{g=1}^{G} \left[\ln|\mathbf{C}| + \left(\boldsymbol{y}^{(g)}\right)^{\mathrm{H}} \mathbf{C}^{-1} \boldsymbol{y}^{(g)} \right] \\ &\qquad - \sum_{k=1}^{K_{\max}} \left(z_k \ln\zeta + (1 - z_k)\ln(1-\zeta) \right) + \text{const.},\end{aligned}$$

where $p(\boldsymbol{y}^{(g)}|\boldsymbol{z}, \boldsymbol{\theta}; \beta, \boldsymbol{\gamma}) = \mathrm{CN}(\boldsymbol{y}^{(g)}; \mathbf{0}, \mathbf{C})$ with $\mathbf{C}$ as in (7). The posterior probabilities of the coefficient vectors $\boldsymbol{\alpha}^{(g)}$, $g = 1, \ldots, G$, are given by (9) with $\boldsymbol{y}$ and $\boldsymbol{\alpha}$ replaced by $\boldsymbol{y}^{(g)}$ and $\boldsymbol{\alpha}^{(g)}$.

The procedure to estimate the variables $\boldsymbol{\theta}$, $\boldsymbol{z}$, $\boldsymbol{\gamma}$, β and ζ follows straightforwardly from the method used in the SMV case. Here we provide a brief discussion of the derivation of the update equations; refer to Sec. II for details.

To estimate $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ the first- and second-order derivatives of $\mathcal{L}_{\text{MMV}}$ are needed. Denote the derivative (12) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$ as $\frac{\partial \mathcal{L}^{(g)}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k}$. Then we have

$$\frac{\partial \mathcal{L}_{\text{MMV}}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k} = \sum_{g=1}^{G} \frac{\partial \mathcal{L}^{(g)}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k}.$$

[9]It is worth noting that in array processing the complete data case corresponds to the very common situation of using a uniform linear array.

A similar result follows for the second-order derivative and the derivatives with respect to $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$.

The estimate of ζ is unchanged from the SMV case (23).

To estimate the noise variance β, we write an upper bound of the same form as (25) and find its minimizer to be

$$\begin{aligned}\hat{\beta} = \max\Big(&\varepsilon_\beta, M^{-1} \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\right) \\ &+ (GM)^{-1} \sum_{g=1}^{G} ||\boldsymbol{y}^{(g)} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \hat{\boldsymbol{\mu}}^{(g)}||^2 \Big),\end{aligned}$$

where $\hat{\boldsymbol{\mu}}^{(g)}$ is given by (10) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$.

To write the activation and deactivation criteria for the MMV model we rewrite the objective in terms of the parameters of a single component, analogously to (27):

$$\begin{aligned}\mathcal{L}_{\text{MMV}}(z_k, \theta_k, \gamma_k) = z_k \Bigg(&G \ln\big(1 + \gamma_k s_{\sim k}(\theta_k)\big) \\ &- \sum_{g=1}^{G} \frac{|q_{\sim k}^{(g)}(\theta_k)|^2}{\gamma_k^{-1} + s_{\sim k}(\theta_k)} + \ln\left(\frac{1-\hat{\zeta}}{\hat{\zeta}}\right) \Bigg) + \text{const.},\end{aligned} \tag{49}$$

where $q_{\sim k}^{(g)}(\theta_k)$ is given by (28) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$. We omit the details of the activation and deactivation stages, as they follow straightforwardly from (49) and the description in Secs. II-C4 and II-C5.

The insightful reader may have noticed that the calculations required for MMV are very similar to those required for SVM. In particular, the matrix $\hat{\mathbf{C}}$ is unchanged and the methods for calculating matrix-vector products involving $\hat{\mathbf{C}}^{-1}$ presented in Secs. III and IV can be utilized. All expressions involving $\boldsymbol{y}$ (i.e., $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{u}$, $\boldsymbol{q}^{\mathcal{G}}$ and $\boldsymbol{y}\hat{\mathbf{C}}^{-1}\boldsymbol{y}$) must be calculated for each observation vector $\boldsymbol{y}^{(g)}$. This means that in the case of complete observations, the generalized Schur algorithm can be used so that the MMV algorithm has per-iteration complexity $\mathcal{O}(N \log^2 N + GN \log N)$. With incomplete observations the semifast method can be used with per-iteration complexity $\mathcal{O}(N\hat{K}^2 + GN \log N)$.

VI. Experiments

A. Setup, Algorithms & Metrics

In our experiments we use the signal model (1). In the following the wrap-around distance on $[0, 1)$ is used for all differences of frequencies (see Footnote 5). Unless otherwise noted, the true frequencies are drawn randomly, such that the minimum separation between any two frequencies is $2/N$. Specifically, the frequencies are generated sequentially for $k = 1, \ldots, K$ with the kth frequency, $\tilde{\theta}_k$, drawn from a uniform distribution on the set $\{\theta \in [0, 1) : d(\theta, \tilde{\theta}_l) > 2/N \text{ for all } l < k\}$.

The true coefficients in $\tilde{\boldsymbol{\alpha}}$ are generated i.i.d. random, with each entry drawn as follows. First a circularly-symmetric complex normal random variable a_k with standard deviation 0.8 is drawn. The coefficient is then found as $\tilde{\alpha}_k = a_k + 0.2\, e^{j \arg(a_k)}$. The resulting random variable has the property $|\tilde{\alpha}_k| \geq 0.2$, i.e., all components have significant magnitude. We use this specification to ensure that all components can be distinguished from noise. After generating the set of K

frequencies and coefficients, the noise variance β is selected such that the desired signal-to-noise ratio (SNR) is obtained, with $\text{SNR} \triangleq ||\boldsymbol{\Phi}\boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}||^2/(M\beta)$.

We compare the superfast LSE algorithm[10] with the following reference algorithms: variational Bayesian line spectral estimation (VALSE) [16]; atomic soft thresholding[11] (AST) [12]; gridless SPICE[12] (GLS) [35]; ESPRIT [6], [8]; and a gridded solution obtained with the least absolute shrinkage and selection operator (LASSO) solved using SpaRSA[13] [36].

The solution to the primal problem of AST [12] directly provides an estimate of the signal vector $\boldsymbol{h} = \boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}$. This solution is known to be biased towards the all-zero solution (as is also the case with the classical LASSO solution). A so-called *debiased* solution can be obtained by recovering the frequencies from the AST dual and estimating the coefficients $\tilde{\boldsymbol{\alpha}}$ via least-squares. As in [12], we report here the debiased solution. If the frequencies are separated by at least $2/N$, the AST algorithm is known to exactly recover the frequencies in the noise-free case [11]–[13]. In the noisy case no such recovery guarantee exists, but a bound on the estimation error of the signal vector $\boldsymbol{h}$ is known [12], [13]. Unfortunately this error bound does not apply to the debiased solution we report herein.

We use the variant of GLS [35] which uses SORTE [9] for model order estimation and MUSIC [7] for frequency estimation.

ESPRIT requires an estimate of the signal covariance matrix and of the model order. The former is obtained as the averaged sample covariance matrix computed from the signal vector split into $N/3$ signal vectors of length $2N/N$ using forward-backward smoothing. The model order is estimated with SORTE [9].

The LASSO solution is obtained using a grid of size $8N$. We have observed that no improvement in performance is achieved with a finer grid. The regularization parameter of LASSO is selected as proposed in [12] with knowledge of the true noise variance. We use the debiased solution returned by the SpaRSA solver.

In the evaluation of the signal reconstruction we have also included an oracle estimator (denoted Oracle) which obtains a least squares solution for $\tilde{\boldsymbol{\alpha}}$ with known $\tilde{\boldsymbol{\theta}}$.

Three performance metrics are used: normalized mean-squared error (NMSE) of the reconstructed signal, block success rate (BSR) and component success rate (CSR). The NMSE is defined as

$$\text{NMSE} \triangleq \frac{||\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\hat{\boldsymbol{\alpha}} - \boldsymbol{\Psi}(\tilde{\boldsymbol{\tau}})\tilde{\boldsymbol{\alpha}}||^2}{||\boldsymbol{\Psi}(\tilde{\boldsymbol{\tau}})\tilde{\boldsymbol{\alpha}}||^2}.$$

The BSR is the proportion of Monte Carlo trials in which the frequency vector $\tilde{\boldsymbol{\theta}}$ is successfully recovered. Successful recovery is understood as correct estimation of the model order K and that $||d(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})||_\infty < 0.5/N$. The association of the entries in $\hat{\boldsymbol{\theta}}$ to those in $\tilde{\boldsymbol{\theta}}$ is obtained by using the Hungarian method [37] (also known as Munkres assigment algorithm) minimizing $||d(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})||_2^2$.

[10]We have published our code at github.com/thomaslundgaard/superfast-lse. It is based on our own implementation of the generalized Schur algorithm and the NUFFT available at cims.nyu.edu/cmcl/nufft/nufft.html.

[11]The code is available online at github.com/badrinarayan/astlinespec.

[12]The code has kindly been provided by the authors.

[13]The code is available online at lx.it.pt/∼mtf/SpaRSA.

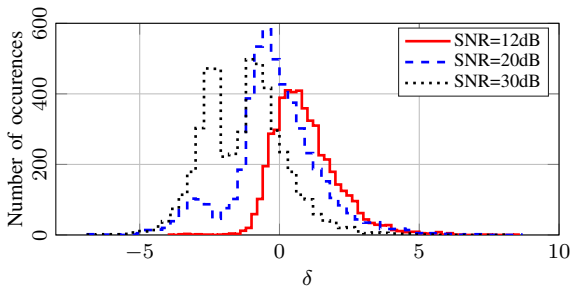

Fig. 1. Histograms of δ values for different signal-to-noise ratios. Cases where $\delta > 0$ correspond to cases where an artefact component is activated if the criterion (33) is used.

The BSR can be misleading, since a trial is considered to be unsuccessful if just a single component is misestimated; for example if a component is represented in the estimate by two components with very close frequencies. We therefore introduce the CSR, defined as follows:

$$\text{CSR} \triangleq \frac{\sum_{k=1}^{\hat{K}} S(\hat{\theta}_k, \tilde{\boldsymbol{\theta}}) + \sum_{k=1}^{K} S(\tilde{\theta}_k, \hat{\boldsymbol{\theta}})}{\hat{K} + K}$$

with the success function $S(x, \boldsymbol{a}) \triangleq \mathbb{1}[\min_k d(x, a_k) < 0.5/N]$, where $\mathbb{1}[\cdot]$ denotes the indicator function. The reported CSR is averaged over a number of Monte Carlo trials. The CSR takes values in $[0, 1]$. A CSR of 1 is achieved if, and only if, all estimated components are in the vicinity of one or more true components and all true components are in the vicinity of one or more estimated components.

B. Choosing the Activation Threshold

To determine a sensible value for the activation threshold τ in (34), the following experiment is conducted. We consider the complete data case with $N = M = 128$ and the number of components is fixed at $K = 35$, as there is a larger tendency to activate artefact components for relatively large K/N. The algorithm is provided with the knowledge of $K_{\max} = 35$ and the activation probability is fixed at $\hat{\zeta} = 35/128$. The algorithm is run with the activation criterion (33). In this way, the algorithm in most cases successfully estimates the frequencies without any artefacts. After the algorithm has terminated, we test if $\tilde{\boldsymbol{\theta}}$ was successfully recovered (as defined above). If so, $K_{\max}$ is increased and the procedure for activating a component in Sec. II-C5 is run and the difference between the left-hand and right-hand sides of (33) is saved. We refer to this difference as δ and the criterion (33) can be expressed as $\delta > 0$. In Fig. 1 we show histograms of the value δ obtained from $5,000$ successful recoveries at three different SNR values At each SNR, the experiment is repeated until the required number of successful recoveries are obtained; trials without successful recovery are discarded. Cases where $\delta > 0$ thus correspond to cases where an artefact would be activated using criterion (33).

The heuristic criterion (34) corresponds to $\delta > \tau$. From Fig. 1 it is clearly seen that threshold $\tau = 5$ is a sensible value, which precludes almost all artefact components from being activated. It is seen that this threshold works well for a

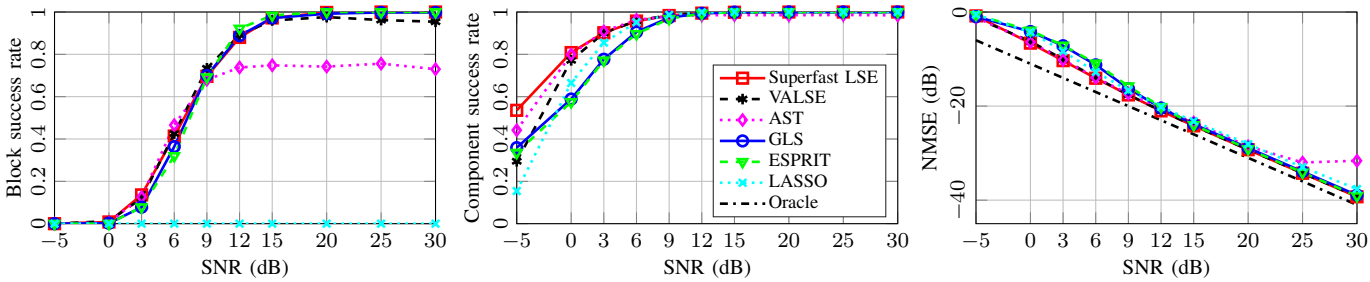

Fig. 2. Simulation results for varying SNR with complete data ($\mathbf{\Phi} = \mathbf{I}$). The signal length is $N = M = 128$ and the number of components is $K = 10$. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

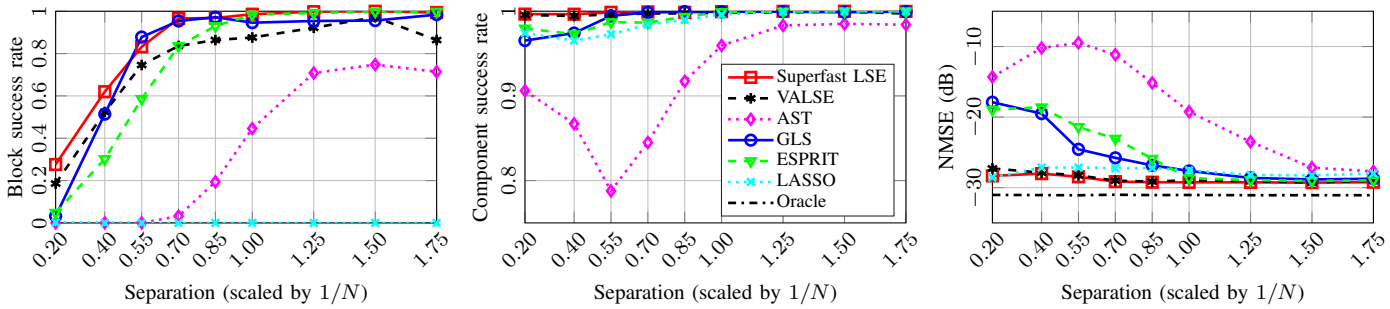

Fig. 3. Simulation results for closely located components with complete data ($\mathbf{\Phi} = \mathbf{I}$). The frequencies are generated as 5 pairs (i.e. $K = 10$) such that each pair has varying (deterministic) intra-pair separation, while the location of the pairs are generated randomly with non-paired frequencies at least $2/N$ apart (i.e., the location of the pairs are generated using a procedure similar to the one described in Sec. VI-A). The signal length is $N = M = 128$ and the SNR is 20 dB. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

large range of SNR values. We have not investigated whether $\tau = 5$ is so large that desired components are precluded from activation. The results in the following investigations are all obtained with $\tau = 5$ and the good performance of our algorithm across this wide selection of scenarios indicates that the selected τ is not too large.

C. Estimation with Complete Data

In Fig. 2 we show performance results versus SNR. We first notice that Superfast LSE is on par with or outperforms all other algorithms in the three metrics shown here for all SNR values. In the low SNR region no algorithm can reliably recover the correct model order and the frequencies. In the plots of the CSR and NMSE, we see that Superfast LSE, VALSE and AST generally achieve the best approximation of the true frequencies. There is a small performance gap in terms of NMSE between Oracle and all other evaluated algorithms due to the uncertainty in frequency estimation (Oracle knows the true frequencies).

ESPRIT and GLS are observed to have the weakest performance at low SNR, especially in terms of CSR and NMSE. Both algorithms use SORTE to estimate the model order from the eigenvalues of the signal covariance. At low SNR it is hard to distinguish the signal eigenvalues from the noise eigenvalues, leading to the observed deterioration in performance.

At medium to high SNR, BSR of AST is about 0.75. The algorithm tends to slightly overestimate the model order (not shown here). We hypothesise that such systematic overestimation of the model order can be avoided by adjusting the regularization parameter used in AST. Doing so would, however, mean that AST would perform worse in other scenarios. This is exactly the weakness of methods involving regularization parameters.

Finally note that LASSO is never able to successfully estimate the model order, due to the use of a grid. In particular each true frequency component is estimated by a few non-zero entries at neighbouring gridpoints. It is visible in the CSR that LASSO indeed estimates frequencies which lie in the vicinity of the true frequencies. In some applications, e.g. channel estimation in wireless communications, it is the reconstructed signal and not the frequencies themself which are of interest. In this case LASSO may be preferable because of its simplicity. Due to the grid approximation, LASSO performs a little worse than the best gridless algorithms in terms of NMSE.

D. Super Resolution

The ability to separate components beyond the Rayleigh limit of $1/N$ is known as super resolution. The results in Fig. 3 illustrate the super resolution ability of the algorithms. In this experiment we generate 5 pairs of frequencies with varying distance between the two frequencies in each pair. The pairs are well separated (at least $2/N$ separation between frequencies which are not in the same pair).

The NMSE performance of Superfast LSE, VALSE and LASSO is only slightly worse at low separation when compared to the performance at large separation. It is evident that the model order and the frequencies cannot be recovered in every case (BSR below 1) when the separation is less than $1/N$. Since the CSR is close to 1 and the NMSE is close to that of Oracle, these three algorithms handle closely located components well, in the sense that a good approximation of the frequencies is obtained.

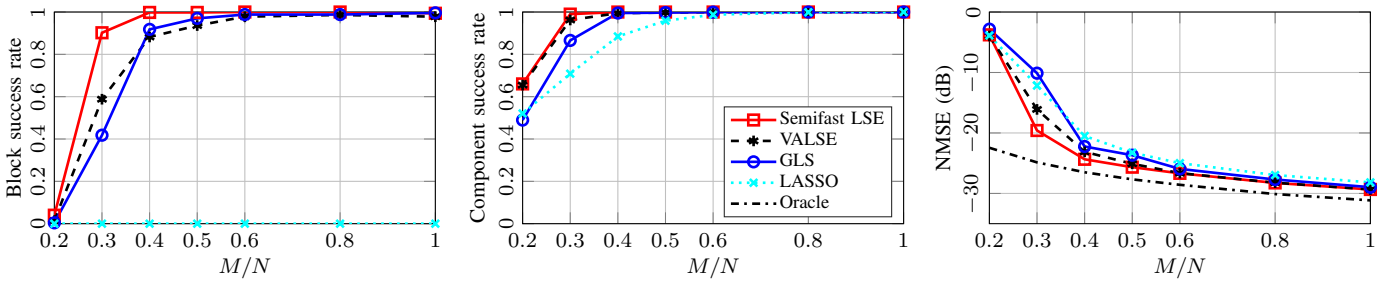

Fig. 4. Simulation results with incomplete data, i.e., $\mathbf{\Phi}$ contains M rows of $\mathbf{I}$ selected at random. The signal length is $N = 128$, the SNR is 20 dB and the number of components is $K = 10$. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

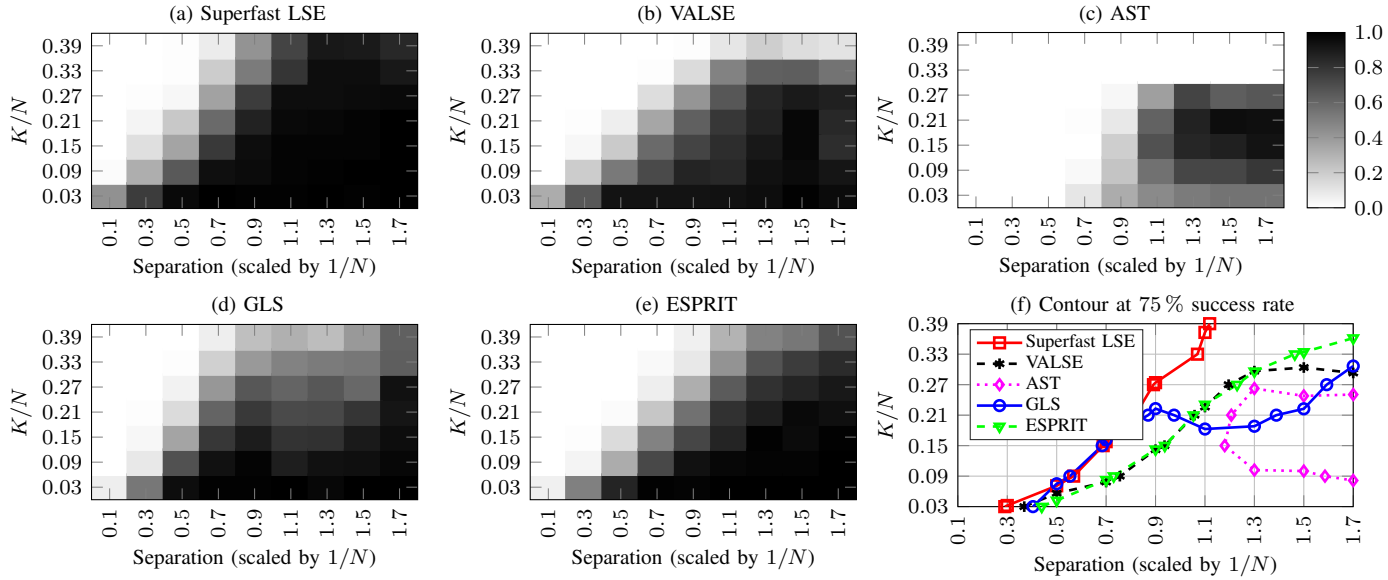

Fig. 5. Simulation results showing phase transitions with complete data ($\mathbf{\Phi} = \mathbf{I}$). The plots show block success rate. The set of frequencies are generated as closely located pairs, following the methodology described in the caption of Fig. 3. The number of pairs are selected such that the specified ratio K/N is achieved as closely as possible. The signal length is $N = M = 128$ and the SNR is 20 dB. Results are averaged over 120 Monte Carlo trials.

AST, GLS and ESPRIT give a CSR below 1 and a rather large NMSE when the separation is small. This is despite the fact that GLS and ESPRIT do not show a significantly worse BSR compared to Superfast LSE. We have observed that this is because these algorithms significantly underestimate the model order in some cases, resulting in a large contribution to NMSE.

ESPRIT shows worse super resolution ability than Superfast LSE, VALSE and GLS (lower BSR for separation below $0.7/N$). This is because a covariance matrix of size $2N/3$ is formed, thus reducing the effective signal length.

E. Estimation with Incomplete Data

Fig. 4 reports the performance in the incomplete data case. The measurement matrix $\mathbf{\Phi}$ is generated by randomly selecting M rows of the $N \times N$ identity matrix. The set of observation indices is chosen to include the first and last indices, while the remaining $M - 2$ indices are obtained by uniform random sampling without replacement. Only a subset of the algorithms are applicable in this case. Our proposed algorithm is implemented using the techniques described in Sec. IV. We refer to it as Semifast LSE.

Semifast LSE and VALSE largely show the same performance, while GLS has a slightly higher NMSE for $M/N \leq 0.5$. The higher NMSE is caused by a few outliers (less than 1% of the Monte Carlo trials) where GLS significantly underestimates the model order. LASSO is again observed to have reasonable NMSE and CSR while being unable to correctly estimate the set of frequencies (i.e., BSR$= 0$).

F. Phase Transitions

Inspired by the concept of phase transitions in compressed sensing, we perform an experiment which shows a similar phenomenon for LSE. In particular we demonstrate that for each algorithm there is a region in the space of system parameters where it can almost perfectly recover the frequencies and a region where it cannot, with a fairly sharp transition between the two. The results, in terms of BSR, are seen in Fig. 5.

We first note that AST has rather poor performance, which is consistent with the observation in Fig. 2 that its BSR is significantly below 1. Turning our attention to VALSE, GLS and ESPRIT, we see that these algorithms generally do not deal well with a large number of components, in the sense that the BSR is significantly below 1 for $K/N \geq 0.15$. It is seen in Fig. 5f that Superfast LSE has the largest region with high probability of successful recovery (BSR ≥ 0.75).

G. Computation Times

In Fig. 6 and 7 we show algorithm runtimes for varying problem sizes. Our proposed method uses the superfast and

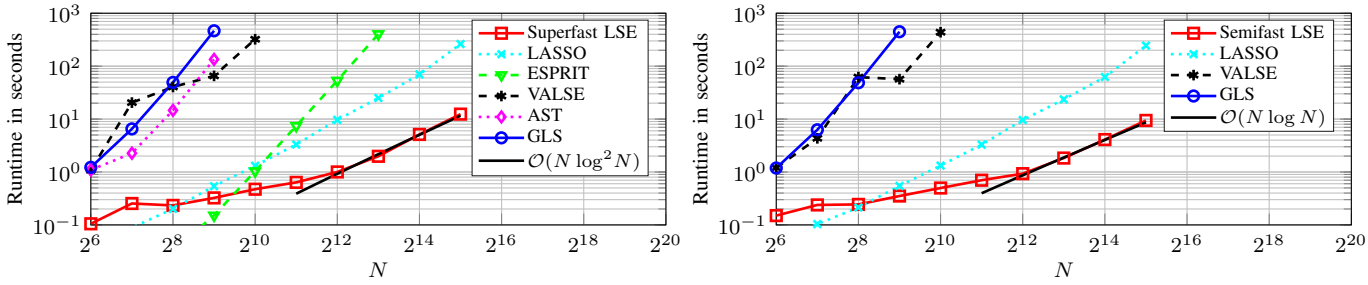

Fig. 6. Runtimes in seconds versus problem size N. We show for both complete (left) and incomplete (right) data case. The number of components is $K = 15$ and the SNR is 20 dB. Values are averaged over 20 Monte Carlo trials. In the incomplete data case we set $M = 0.75N$. We also plot (solid black) the asymptotic per-iteration complexity of Superfast and Semifast LSE.

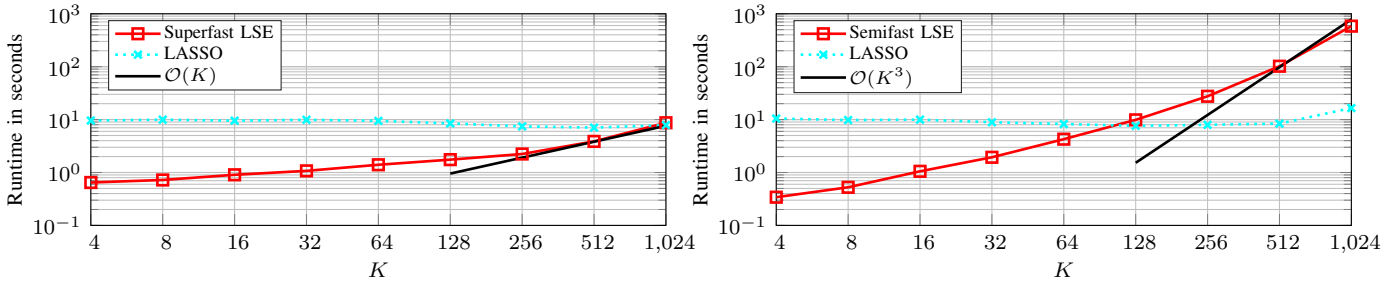

Fig. 7. Runtimes in seconds versus number of components K. We show for both complete (left) and incomplete (right) data case. The problem size is $N = 4096$ and the SNR is 20 dB. Values are averaged over 20 Monte Carlo trials. We only show results for algorithms which has runtime lower than 10 s at $K = 4$. In the incomplete data case we set $M = 0.75N = 3072$. We also plot (solid black) the asymptotic per-iteration complexity of Superfast and Semifast LSE.

semifast implementations in Secs. III and IV, respectively. The results are obtained using MATLAB on a 2011 MacBook Pro. To avoid differences in results originating from the amount of parallelism achieved in each implementation, MATLAB is restricted to only use a single computational thread. The part of the code where each algorithm spends significant time is implemented as native code via MATLAB's codegen feature.

For varying N (Fig. 6), we first observe that for small to moderate problem sizes ($N \leq 2^{10}$) the difference between LASSO and our proposed algorithms is small (less than 1 second). This difference is mainly due to implementation details. In the large-N region, Superfast and Semifast LSE are approximately an order of magnitude faster than LASSO. We observe that the asymptotic per-iteration complexity of Superfast and Semifast LSE describes the scaling of the total runtime well for $N \geq 2^{12}$, because the number of iterations (not shown) stays practically constant. The state-of-the-art LSE methods VALSE, AST and GLS all have $\mathcal{O}(N^3)$ time complexity or worse, which results in very large runtimes even when the problem size is moderate (e.g. > 100 s for AST and GLS at $N = 512$). For large problem sizes, the $\mathcal{O}(N^3)$ time complexity of ESPRIT is evident and Superfast/Semifast LSE and LASSO significantly outperform ESPRIT.

In Fig. 7 we show results illustrating how the computation time scales with K. In this analysis we assume $\hat{K} = \mathcal{O}(K)$. First we note that the runtime of LASSO is practically constant with K. In the complete data case, the per-iteration complexity of Superfast LSE scales linearly with K. In practice we see a slower scaling with K, which means that the values of K we use here are not large enough to reach the asymptotic region. Simulations with large K cannot be run, because we need approximately $K < N/4$ for $\hat{K} = \mathcal{O}(K)$ to hold (cf. Fig. 5).

In the incomplete data case, we see that the runtime of Semifast LSE increases quickly with K, such that for $K > 128$ LASSO is faster than Semifast LSE. We do, however, see that the asymptotic complexity of $\mathcal{O}(K^3)$ is not reached in our experiment, because the runtime is dominated by the calculation of $\boldsymbol{s}^{\mathcal{G}}$, which has complexity $\mathcal{O}(LK^2)$.

VII. Conclusions

We have presented a low-complexity algorithm for line spectral estimation. Computational methods for both the complete and incomplete data cases have been presented, along with an extension to the case of multiple measurement vectors.

The proposed algorithm falls in the category of Bayesian methods for line spectral estimation. Bayesian methods are widely accepted due to their high estimation accuracy, but a drawback of this class of methods has historically been their large computational complexity. In that respect, this work makes an important contribution in making Bayesian methods more viable in practice.

At the core of the computational method for the complete data case lies the application of the Gohberg-Semencul formula to the Toeplitz signal covariance matrix. Many methods for line spectral estimation have Toeplitz covariance matrices at their core and we conjecture that the computational complexity of some of them can be drastically reduced by applying the techniques we have demonstrated in this paper.

Our numerical experiments show that our Superfast LSE algorithm has very high estimation accuracy. For example, in Fig. 5 we see that Superfast LSE attains high frequency recovery rates for a much larger set of scenarios than any of the reference algorithms. At the same time our algorithm has so low computation time that it makes highly-accurate LSE feasible for problems with size much larger than methods currently available in the literature can practically deal with.

APPENDIX A
CONVERGENCE ANALYSIS

We now discuss the convergence of our proposed block-coordinate descent algorithm. To do so we introduce an iteration index i on all estimated variables. Our algorithm then produces sequences of blocks of estimates denoted as $\{\hat{\boldsymbol{z}}^i\}$, $\{\hat{\zeta}^i\}$, $\{\hat{\beta}^i\}$ and $\{(\hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i)\}$. We denote the value of the objective function at the end of the ith iteration as $\mathcal{L}^i \triangleq \mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^i, \hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i)$. We first note that all updates of the estimates are guaranteed not to increase the objective, so $\{\mathcal{L}^i\}$ is a non-increasing sequence. Since $\beta \geq \varepsilon_\beta$ it is also bounded below and thus it converges. Therefore, our proposed algorithm terminates in finite time.

Unfortunately, convergence of $\{\mathcal{L}^i\}$ does not imply convergence of the sequences of estimates. Even with exact minimization in each block of variables, block-coordinate descent on non-convex functions can get stuck in an infinite cycle [38]. This is further complicated by the fact that our algorithm only approximately solves the minimization in some of the blocks.

Proposition 5 in [39] shows that with exact minimization in each block, block-coordinate descent converges if the objective function is strictly quasiconvex in all but 2 blocks. The objective function $\mathcal{L}$ is strictly quasiconvex in ζ and β. There is thus hope that we can prove convergence of our scheme which, in lieu of computing the exact minimizer, merely has a descent property in each block. Our approach to show convergence shares the same overall idea as that in [39], while many details differ.

To discuss the convergence properties, we first derive a number of lemmas. Theorem 1 is proved at the end of the appendix. For notational simplicity we take the convention that for each i the block-coordinate descent algorithm cycles through the block updates in the following order: $\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^i$ and finally $(\hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i)$, such that for example $\hat{\beta}^i$ is found based on $(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1})$. This is strictly speaking not how we have defined our algorithm, but that does not affect the correctness of our analysis.

Lemma 1: The sequence of estimates has at least one convergent subsequence, i.e., at least one limit point.

Proof: All variables but $\boldsymbol{\gamma}$ and β are defined to be in a closed and bounded set. Since $\lim_{\beta\to\infty} \mathcal{L} = \infty$ we can restrict β to a closed and bounded set determined by the (finite) initial value of the objective function. A similar argument holds for each γ_k. The lemma then follows from the Bolzano-Weierstrass theorem. ∎

Lemma 2: The sequence $\{\hat{\boldsymbol{z}}^i\}$ converges.

Proof: Each activation of a component gives a decrease in $\mathcal{L}$ of at least $\varepsilon_{\mathcal{L}}$. Since $\{\mathcal{L}^i\}$ is lower bounded, there can only be finitely many activations. Since there cannot be more deactivations than activations, also the number of deactivations is finite. There is thus a finite number of changes to $\hat{\boldsymbol{z}}$ and $\{\hat{\boldsymbol{z}}^i\}$ converges. We denote the limit point as $\bar{\boldsymbol{z}}$. ∎

Lemma 3: The sequence $\{\hat{\zeta}^i\}$ converges. Further, the limit point $\bar{\zeta}$ is the unique global minimizer of $\zeta \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \zeta, \beta, \boldsymbol{\theta}, \boldsymbol{\gamma})$ for any β, $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$.

Proof: The first statement follows from Lemma 2 since $\hat{\zeta}^i$ (23) is only a function of $\hat{\boldsymbol{z}}^i$. The second statement results from the fact that $\hat{\zeta}^i$ is defined as the global minimizer of $\zeta \mapsto \mathcal{L}(\hat{\boldsymbol{z}}^i, \zeta, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1})$ and this global minimizer does not depend on $\hat{\beta}^{i-1}$, $\hat{\boldsymbol{\theta}}^{i-1}$ and $\hat{\boldsymbol{\gamma}}^{i-1}$. ∎

Lemma 4: The sequence $\{\hat{\beta}^i\}$ converges to the limit point $\bar{\beta}$. Further, for every limit point $(\bar{\boldsymbol{z}}, \bar{\zeta}, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})$ of the remaining variables, the limit point $\bar{\beta}$ is a local minimum at the boundary ε_β or a stationary point of $\beta \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \bar{\zeta}, \beta, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})$.

Proof: To perform this proof we expand our previous notation and denote the upper bound (25) as $Q(\beta; \hat{\boldsymbol{z}}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1})$. Then,

$$\begin{aligned}\mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) &= Q(\hat{\beta}^{i-1}; \hat{\boldsymbol{z}}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \\ &\geq Q(\hat{\beta}^i; \hat{\boldsymbol{z}}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \geq \mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^i, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}).\end{aligned}$$

Recalling that $\{\mathcal{L}^i\}$ converges, we have

$$\begin{aligned}\lim_{i\to\infty} \Big| &\mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \\ &- \mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\zeta}^i, \hat{\beta}^i, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \Big| = 0,\end{aligned}$$

and thus

$$\begin{aligned}\lim_{i\to\infty} \Big| &Q(\hat{\beta}^{i-1}; \hat{\boldsymbol{z}}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \\ &- Q(\hat{\beta}^i; \hat{\boldsymbol{z}}^i, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\boldsymbol{\gamma}}^{i-1}) \Big| = 0. \quad (50)\end{aligned}$$

Reasoning by contradiction, assume that the sequence of estimates $\{\hat{\beta}^i\}$ has two limit points $\bar{\beta}_1$ and $\bar{\beta}_2$, such that $\bar{\beta}_1 \neq \bar{\beta}_2$. Let $(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})$ be any limit point of $\{\hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i\}$ (such a limit point exists due to Lemma 1). Then by (50) we must have

$$Q(\bar{\beta}_1; \bar{\boldsymbol{z}}, \bar{\beta}_1, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}}) = Q(\bar{\beta}_2; \bar{\boldsymbol{z}}, \bar{\beta}_1, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}}). \quad (51)$$

Recalling the definition of $\hat{\beta}^i$, we have that $\bar{\beta}_2$ uniquely minimizes Q. Then, since we assumed $\bar{\beta}_1 \neq \bar{\beta}_2$, we get

$$Q(\bar{\beta}_1; \bar{\boldsymbol{z}}, \bar{\beta}_1, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}}) > Q(\bar{\beta}_2; \bar{\boldsymbol{z}}, \bar{\beta}_1, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}}),$$

which contradicts (51). So $\{\hat{\beta}^i\}$ has only a single limit point which we denote as $\bar{\beta}$.

To prove the second statement, use (24) to show that

$$\frac{\partial}{\partial\beta} Q(\beta; \bar{\boldsymbol{z}}, \bar{\beta}, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})\Big|_{\beta=\bar{\beta}} = \frac{\partial}{\partial\beta} \mathcal{L}(\bar{\boldsymbol{z}}, \bar{\zeta}, \beta, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})\Big|_{\beta=\bar{\beta}}.$$

If $\bar{\beta} = \varepsilon_\beta$, we have that the derivatives of Q and thus of $\mathcal{L}$ are positive. It follows that $\bar{\beta}$ is a local or global minimum. If $\bar{\beta} \neq \varepsilon_\beta$ it is, by definition, a stationary point of Q. It is therefore also a stationary point of $\beta \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \bar{\zeta}, \beta, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\gamma}})$. ∎

We can now give a proof of the main convergence result.

Proof of Theorem 1: Convergence to a unique limit follows immediately from the assumption and Lemmas 2, 3 and 4.

To prove the second statement, we first note that $\mathcal{L}$ is constant with respect to those entries of $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ for which $\bar{z}_k = 0$. It then follows from the assumption that $\frac{\partial\mathcal{L}}{\partial\theta_k} = 0$ and $\frac{\partial\mathcal{L}}{\partial\gamma_k} = 0$ for all $k = 1, \ldots, K_{\max}$ at the limit point. Similarly from Lemma 3 we have $\frac{\partial\mathcal{L}}{\partial\zeta} = 0$ at the limit point. If $\bar{\beta} \neq \varepsilon_\beta$ we have $\frac{\partial\mathcal{L}}{\partial\beta} = 0$ at the limit point and the result follows immediately.

If $\bar{\beta} = \varepsilon_\beta$ the result can be obtained by introducing a

Lagrange multiplier such that the limit point satisfies the Karush-Kuhn-Tucker conditions. ■

APPENDIX B
EFFICIENT EVALUATION OF $\omega_s(i)$, $\omega_t(i)$, $\omega_v(i)$ AND $\omega_x(i)$

We derive a low-complexity computation of $\omega_s(i)$ by first inserting (35) into (39) to get

$$\omega_s(i) = \delta_{N-1}^{-1}\left(\sum_{q=\max(0,-i)}^{\min(N-1,N-1-i)}\sum_{r=0}^{q} -\rho_{q-r-1}\rho_{q+i-r-1}^{*} + \rho_{N-1+r-q}^{*}\rho_{N-1+r-q-i}\right) \quad (52)$$

for $i = -(N-1),\ldots,N-1$. Then note that since $\mathbf{C}$ is Hermitian we have $\omega_s(-i) = \omega_s^*(i)$. We therefore restrict our attention to $i \geq 0$ in the following. We need the identity

$$\sum_{q=0}^{N-1}\sum_{r=0}^{q} z_{q,r} = \sum_{q=0}^{N-1}\sum_{k=0}^{N-1-q} z_{q+k,k}, \quad (53)$$

from which we get (recall that $\rho_i = 0$ for $i < 0$ and $i > N-1$)

$$\omega_s(i) = \delta_{N-1}^{-1}\sum_{q=0}^{N-1-i}(N-i-q)(\rho_{N-1-q}^{*}\rho_{N-1-q-i} - \rho_{q-1}\rho_{q-1+i}^{*}).$$

Substituting $q = N-1-\bar{q}-i$ in the first term and $q = \bar{q}+1$ in the second term we finally obtain

$$\omega_s(i) = \delta_{N-1}^{-1}\sum_{\bar{q}=0}^{N-1} c_s(\bar{q},i)\rho_{\bar{q}}\rho_{\bar{q}+i}^{*}, \quad (54)$$

where $c_s(\bar{q},i) \triangleq (2-N+i+2\bar{q})$. The above expression can be calculated as the sum of two cross-correlations in time $\mathcal{O}(N\log N)$ by using FFT techniques.

To evaluate $\omega_t(i)$ (40) we again insert (35) and get

$$\omega_t(i) = \delta_{N-1}^{-1}\left(\sum_{q=\max(0,-i)}^{\min(N-1,N-1-i)} 2\pi q\sum_{r=0}^{q} -\rho_{q-r-1}\rho_{q+i-r-1}^{*} + \rho_{N-1+r-q}^{*}\rho_{N-1+r-q-i}\right) \quad (55)$$

for $i = -(N-1),\ldots,N-1$. Be aware that we do not have $\omega_t(i) = \omega_t^*(-i)$. Applying (53), performing the same substitutions as above and following tedious, but straightforward, algebra we finally get

$$\omega_t(i) = \frac{2\pi}{\delta_{N-1}}\sum_{\bar{q}=0}^{N-1} c_t(\bar{q},i)\rho_{\bar{q}}\rho_{\bar{q}+i}^{*} \quad (56)$$

with

$$c_t(\bar{q},i) \triangleq -\bar{q}(\bar{q}+i) + i\left(N-\frac{3+i}{2}\right) + \bar{q}^2 + (N-1)\left(\bar{q}-\frac{N-2}{2}\right), \quad (57)$$

which again can be evaluated using FFT techniques.

Omitting details, we use a similar approach to find

$$\omega_v(i) = \frac{4\pi^2}{\delta_{N-1}}\sum_{\bar{q}=0}^{N-1} c_v(\bar{q},i)\rho_{\bar{q}}\rho_{\bar{q}+i}^{*} \quad (58)$$

$$\omega_x(i) = \frac{4\pi^2}{\delta_{N-1}}\sum_{\bar{q}=0}^{N-1} c_x(\bar{q},i)\rho_{\bar{q}}\rho_{\bar{q}+i}^{*}, \quad (59)$$

where $\omega_v(-i) = \omega_v^*(i)$. The expression giving $\omega_v(i)$ is valid for $i \geq 0$, while that giving $\omega_x(i)$ is valid for $i = -(N-1),\ldots,N-1$. We have also defined

$$c_v(\bar{q},i) = \bar{q}(\bar{q}+i)^2 + (3\bar{q}-2N\bar{q}-\bar{q}^2)(\bar{q}+i) + \frac{2}{3}\bar{q}^3 + (N-1)\bar{q}^2 + \left(N^2-3N+\frac{7}{3}\right)\bar{q} + \frac{3}{2}(i-N)^2 + \frac{1}{3}(i^3-N^3) + \left(\frac{13}{6}-Mi\right)(i-N) + 1$$

$$c_x(\bar{q},i) = (\bar{q}^2+2\bar{q}-N\bar{q})(\bar{q}+i) - \frac{1}{3}\bar{q}^3 + \left(N^2-3N+\frac{7}{3}\right)\bar{q} - \frac{1}{6}i^3 + \left(\frac{3N^2-9N+7}{6}\right)i - \frac{1}{3}N^3 + \frac{3}{2}N^2 - \frac{13}{6}N + 1.$$

REFERENCES

[1] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug 2005.

[2] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Process.*, vol. 40, pp. 590–600, Mar. 1992.

[3] R. Carriere and R. L. Moses, "High resolution radar target modeling using a modified Prony estimator," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 13–18, Jan. 1992.

[4] W. Bajwa, A. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, Jun. 2010.

[5] X. Andrade, J. N. Sanders, and A. Aspuru-Guzik, "Application of compressed sensing to the simulation of atomic systems," *Proc. Nat. Academy of Sciences*, vol. 109, pp. 13 928–13 933, Jul. 2012.

[6] S.-Y. Kung, K. S. Arun, and B. D. Rao, "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem," *J. of the Optical Soc. of America*, vol. 73, pp. 1799–1811, Dec. 1983.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, pp. 276–280, Mar. 1986.

[8] R. Roy and T. Kailath, "ESPRIT - estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, pp. 984–995, Jul. 1989.

[9] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 2006–2021, Nov 2010.

[10] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, pp. 3809–3822, Jul. 2012.

[11] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Trans. Inf. Theory*, vol. 59, pp. 7465–7490, Nov. 2013.

[12] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, pp. 5987–5999, Dec. 2013.

[13] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *J. of Fourier Anal. and Applicat.*, vol. 19, pp. 1229–1254, Dec. 2013.

[14] F. Dublanchet, J. Idier, and P. Duwaut, "Direction-of-arrival and frequency estimation using Poisson-Gaussian modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, vol. 5, Apr. 1997, pp. 3501–3504.

[15] T. L. Hansen, M.-A. Badiu, B. H. Fleury, and B. D. Rao, "A sparse Bayesian learning algorithm with dictionary parameter estimation," in *Proc. 8th IEEE Sensor Array and Multichannel Signal Process. Workshop*, Jun. 2014, pp. 385–388.
[16] M. A. Badiu, T. L. Hansen, and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Trans. Signal Process.*, vol. 65, pp. 2247–2261, May 2017.
[17] G. S. Ammar and W. B. Gragg, "The generalized Schur algorithm for the superfast solution of Toeplitz systems," in *Rational approximation and its applications in mathematics and physics*, 1987, pp. 315–330.
[18] ——, "Numerical experience with a superfast real Toeplitz solver," *Linear Algebra and its Applicat.*, vol. 121, pp. 185–206, Aug. 1989.
[19] B. R. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, pp. 1333–1335, Oct. 1985.
[20] I. Gohberg and I. A. Feldman, *Convolution Equations and Projection Methods for Their Solution*, ser. Translations of mathematical monographs. American Mat. Soc., 2005, vol. 41.
[21] L. Greengard and J.-Y. Lee, "Accelerating the nonuniform fast Fourier transform," *SIAM Review*, p. 443–454, 2004.
[22] J.-Y. Lee and L. Greengard, "The nonuniform FFT of Type 3 and its applications," *J. Comput. Phys.*, pp. 1–5, 2005.
[23] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation," *IEEE Trans. Signal Process.*, vol. 44, pp. 2988–2998, Dec. 1996.
[24] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. 9th Int. Workshop Artificial Intell. and Stat.*, Jan. 2003, pp. 3–6.
[25] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, pp. 2153–2164, Aug. 2004.
[26] J. J. Kormylo and J. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 28, pp. 482–488, May 1982.
[27] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. Springer, 2006.
[28] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.
[29] F. Dellaert, "The expectation maximization algorithm," Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20, Feb. 2002.
[30] D. Shutin, B. H. Fleury, and N. Schneckenburger, "Artifact suppression for super-resolution sparse Bayesian learning," 2018, submitted to *IEEE Trans. Signal Process.*
[31] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, "Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals," *IEEE Trans. Signal Process.*, vol. 59, pp. 6257–6261, Dec. 2011.
[32] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, pp. 1190–1208, 1995.
[33] W. F. Mascarenhas, "The BFGS method with exact line searches fails for non-convex objective functions," *Math. Programming*, vol. 99, pp. 49–61, Jan. 2004.
[34] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, pp. 2477–2488, Jul. 2005.
[35] Z. Yang and L. Xie, "On gridless sparse methods for line spectral estimation from complete and incomplete data," *IEEE Trans. Signal Process.*, vol. 63, pp. 3139–3153, Jun. 2015.
[36] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, pp. 2479–2493, Jul. 2009.
[37] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, pp. 83–97, 1955.
[38] M. J. D. Powell, "On search directions for minimization algorithms," *Math. Programming*, vol. 4, pp. 193–201, Dec. 1973.
[39] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations research letters*, vol. 26, pp. 127–136, Apr. 2000.

Thomas Lundgaard Hansen received the B.Sc. and M.Sc. (cum laude) in electrical engineering from Aalborg University, Denmark in 2011 and 2014, respectively. Since 2014 he has been pursuing the Ph.D. degree in wireless communication at Aalborg University. During 2013 and 2015 he was a visiting scholar at the University of California, San Diego, USA. He is the recipient of the best student paper award (1st place) at the 2014 IEEE Sensor Array and Multichannel Signal Processing workshop and also received an award from IDA Efondet for his master's thesis. His research interests include signal processing, machine learning, optimization and wireless communication.

Bernard Henri Fleury (M'97–SM'99) received the Diplomas in Electrical Engineering and in Mathematics in 1978 and 1990 respectively and the Ph.D. Degree in Electrical Engineering in 1990 from the Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland. Since 1997, he has been with the Department of Electronic Systems, Aalborg University, Denmark, as a Professor of Communication Theory. From 2000 till 2014 he was Head of Section, first of the Digital Signal Processing Section and later of the Navigation and Communications Section. From 2006 to 2009, he was partly affiliated as a Key Researcher with the Telecommunications Research Center Vienna (ftw.), Austria. Prof. Fleury's research interests cover numerous aspects within communication theory, signal processing, and machine learning, mainly for wireless communication systems and networks.

Bhaskar D. Rao (S'80–M'83–SM'91–F'00) is currently a Distinguished Professor in the Electrical and Computer Engineering department and the holder of the Ericsson endowed chair in Wireless Access Networks at the University of California, San Diego. Prof. Rao was elected fellow of IEEE in 2000 and is the recipient of the 2016 IEEE Signal Processing Society Technical Achievement Award. Prof. Rao's interests are in the areas of digital signal processing, estimation theory, and optimization theory, with applications to digital communications, speech signal processing, and biomedical signal processing.