Computationally Efficient Sparsity-Inducing Coherence Spectrum Estimation of Complete and Non-Complete Data Sets

Angelopoulos, Kostas; Glentis, George-Othan; Jakobsson, Andreas

Published in:
Signal Processing

DOI:
10.1016/j.sigpro.2012.12.003

2013

LUND UNIVERSITY

Computationally Efficient Sparsity-Inducing Coherence Spectrum Estimation of Complete and Non-Complete Data Sets

K. ANGELOPOULOS, G. O. GLENTIS, AND A. JAKOBSSON

Published in: Elsevier Signal Processing

doi:10.1016/j.sigpro.2012.12.003

Lund 2013

Mathematical Statistics

Centre for Mathematical Sciences

Lund University

Computationally Efficient Sparsity-Inducing Coherence Spectrum Estimation of Complete and Non-Complete Data Sets ☆

K. Angelopoulos[a], G. O. Glentis[*,a], A. Jakobsson[b]

[a] *Department of Science and Technology of Telecommunications, University of Peloponnese, Tripolis, 22100 Greece*

[b] *Department of Mathematical Statistics, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden*

Abstract

The magnitude squared coherence (MSC) spectrum is an often used frequency-dependent measure for the linear dependency between two stationary processes, and the recent literature contain several contributions on how to form high-resolution data-dependent and adaptive MSC estimators, and on the efficient implementation of such estimators. In this work, we further this development with the presentation of computationally efficient implementations of the recent iterative adaptive approach (IAA) estimator, present a novel sparse learning via iterative minimization (SLIM) algorithm, discuss extensions to two-dimensional data sets, examining both the case of complete data sets and when some of the observations are missing. The algorithms further the recent development of exploiting the estimators' inherently low displacement rank of the necessary products of Toeplitz-like matrices, extending these formulations to the coherence estimation using IAA and SLIM formulations. The performance of the proposed algorithms and implementations are illustrated both with theoretical complexity measures and with numerical simulations.

Key words: Coherence spectrum, data adaptive estimators, efficient algorithms, sparse estimators

☆This work was supported in part by the Swedish Research Council and Carl Trygger's foundation. Parts of this material has been published at the IEEE DSP 2011 workshop.

*Corresponding author. Phone: +30-2710372202. Fax: +30-2710372161.

Email addresses: kaggelop@uop.gr (K. Angelopoulos), gglentis@uop.gr (G. O. Glentis), aj@maths.lth.se (A. Jakobsson)

1. Introduction

The problem of estimating the magnitude squared coherence (MSC) between two or more measured signals is frequently occurring in a wide variety of fields, such as speech processing, time series analysis, geophysics, biomedical engineering, and synthetic aperture radar imaging, wherein one wishes to determine the linear relationship between signals or to determine if a common signal is present in several different measurements. Recently, non-parametric data-adaptive estimation techniques have been exploited to form robust high-resolution MSC estimates [1–5]. In [1, 2], it was shown that the one- and two-dimensional (2-D) Capon and APES-based MSC estimators allow for high-resolution MSC estimates, by forming data-adaptive filter banks, with each filter being constrained to pass its center frequency undistorted while suppressing the contribution of all other components. In [3], this work was extended to allow for non-uniformly sampled data by exploiting a formulation based on the recent iterative adaptive approach (IAA) [6]. The IAA-based MSC algorithm, as well as a segmented version termed SIAA-MSC, was there shown to yield reliable estimates even if a large proportion of the measurements are missing. In this paper, we further extend these works by a proposing 1-D and 2-D formulations of the IAA-based MSC estimator, as well as for a novel semi-parametric SLIM-based estimator. The sparse learning via iterative minimization (SLIM) method was introduced in the context of MIMO radar imaging in [7], and can be viewed as a version of the well-known (regularized) FOCUSS algorithm [8], although including also the iterative estimation of the noise variance (see also [9]). Both the IAA and SLIM algorithms have been shown to converge locally [7, 10], as well as to yield excellent performance for both complete or incomplete data sets. Regrettably, both algorithms are also computationally cumbersome, and several works have focused on forming various computationally efficient implementations for uniformly and non-uniformly sampled data sequences [11–17]. The presented work may be viewed as a continuation of our recently proposed efficient implementation of the Capon- and APES-based MSC estimators [18], wherein we combined earlier efforts in forming computationally efficient implementations of these spectral estimators [19–26] with the inherently low displacement rank of the estimators' products of Toeplitz-like matrices, thereby allowing for the development of appropriate Gohberg-Semencul (GS) representations of these matrices. The resulting implementation

was found to be several orders of magnitude more efficient than the straightforward implementations. Here, building on this work, we extend the IAA- and proposed SLIM-based MSC estimators in a similar way. The paper is organized as follows: in the following section, we briefly review data-adaptive MSC estimation, comparing the formulations of the earlier introduced Capon-, APES-, and IAA-based MSC estimators, as well as introduce a novel SLIM-based MSC estimator. Then, in Section 3, we recall formulations of the MSC estimators using trigonometric polynomials, and then, in Section 4, introduce the efficient implementations of the IAA- and SLIM-based estimators using appropriate GS representations for the necessary products of Toeplitz-like matrices. In Section 5, we proceed to discuss the case of incomplete data sets, followed by the extensions to 2-D formulations of the estimators in Section 6. Section 7 contains a study of the performance of the discussed estimators and implementations. Finally, we conclude on the work in Section 8.

2. Data-Adaptive MSC Estimation

The MSC spectrum, $\gamma^2_{x_1x_2}(\omega)$, of two stationary complex valued signals, $x_1(n)$ and $x_2(n)$, for $n = 0, 1, \ldots, N-1$, is defined as (see, e.g., [27–29])

$$\gamma^2_{x_1x_2}(\omega) = \frac{|S_{x_1x_2}(\omega)|^2}{S_{x_1}(\omega)S_{x_2}(\omega)} \tag{1}$$

where $S_{x_1}(\omega)$ and $S_{x_2}(\omega)$ denote the (auto) spectra of the signals $x_1(n)$ and $x_2(n)$, respectively, whereas $S_{x_1x_2}(\omega)$ denotes the cross-spectrum between these two signals. The Capon-, APES-, and IAA-based MSC estimates are formed using the matched filter bank framework (see also [29, 30]). Let $\mathbf{h}_N^{(i)} \in C^{N\times 1}$ denote a narrowband data dependent finite impulse response (FIR) filter centered at a generic frequency $\omega \in (-\pi, \pi]$, and form the signals of interest into $N \times 1$ subvectors

$$\mathbf{x}_N^{(i)} = \left[\begin{array}{cccc} x_i(0) & x_i(1) & \ldots & x_i(N-1) \end{array} \right]^T \tag{2}$$

where $i = 1$ or 2 for the respective signal, and where $(\cdot)^T$ denotes the transpose. As the filters are narrowband, aiming to only pass the generic frequency ω undistorted whereas the contribution from all other frequencies are minimized, the matched filter bank spectral estimate at frequency ω is found as the power of the filtered signal, i.e.,

$$S_{x_i}(\omega) \approx \mathbf{h}_N^{(i)H} \boldsymbol{\mathcal{R}}_N^{(i)} \mathbf{h}_N^{(i)} \tag{3}$$

where $\boldsymbol{\mathcal{R}}_N^{(i)}$ represents the signal's covariance matrix, defined as

$$\boldsymbol{\mathcal{R}}_N^{(i)} = E\left\{\mathbf{x}_N^{(i)}\mathbf{x}_N^{(i)H}\right\} \tag{4}$$

with $i = 1$ or 2 for the respective signal, where $E\{\cdot\}$ denote the expectation and $(\cdot)^H$ the conjugate transpose, respectively, and where $\mathbf{h}_N^{(i)}$ is a data dependent narrow band filter formed such that

$$\mathbf{h}_N^{(i)} = \text{argmin}_{\mathbf{h}_N^{(i)}} S_{x_i}(\omega) \quad \text{subj. to} \quad \mathbf{h}_N^{(i)H}\mathbf{f}_N(\omega) = 1 \tag{5}$$

where

$$\mathbf{f}_N(\omega) = \left[\begin{array}{cccc} 1 & e^{\jmath\omega} & \ldots & e^{\jmath(N-1)\omega} \end{array}\right]^T \tag{6}$$

is the frequency steering vector. Minimization of (5) with respect to the unknown parameters vector results in a data adaptive and frequency dependent optimal filter of the form

$$\mathbf{h}_N^{(i)} = \frac{[\boldsymbol{\mathcal{R}}_N^{(i)}]^{-1}\mathbf{f}_N(\omega)}{\mathbf{f}_N^H(\omega)[\boldsymbol{\mathcal{R}}_N^{(i)}]^{-1}\mathbf{f}_N(\omega)} \tag{7}$$

The cross-spectral density needed to form (1) is estimated as

$$S_{x_1x_2}(\omega) \approx \mathbf{h}_N^{(1)H}\boldsymbol{\mathcal{R}}_N^{(12)}\mathbf{h}_N^{(2)} \tag{8}$$

with $\boldsymbol{\mathcal{R}}_N^{(12)}$ denoting the cross-covariance matrix, defined as

$$\boldsymbol{\mathcal{R}}_N^{(12)} = E\left\{\mathbf{x}_N^{(1)}\mathbf{x}_N^{(2)H}\right\} \tag{9}$$

Combining (1), (3), (7), and (8), one obtains the Capon-based MSC estimator [1, 4]

$$\gamma_{x_1x_2}^2(\omega) = \frac{\left|\mathbf{f}_N^H(\omega)\boldsymbol{\mathcal{P}}_N^{(12)}\mathbf{f}_N(\omega)\right|^2}{\prod_{i=1}^2 \mathbf{f}_N^H(\omega)[\boldsymbol{\mathcal{R}}_N^{(i)}]^{-1}\mathbf{f}_N(\omega)} \tag{10}$$

where

$$\boldsymbol{\mathcal{P}}_N \triangleq [\boldsymbol{\mathcal{R}}_N^{(1)}]^{-1}\boldsymbol{\mathcal{R}}_N^{(12)}[\boldsymbol{\mathcal{R}}_N^{(2)}]^{-1} \tag{11}$$

2.1. IAA-based MSC estimation

As shown in [1, 2], the Capon- and APES-based MSC estimates result from two different design choices for the narrowband filters and use the standard time based averages approximation in place of estimates of the auto and cross correlation matrices. The IAA-based

algorithm instead forms the covariance matrices as the sum of the spectral contribution from all possible frequency grid points, essentially viewing that data as a sum of sinusoids, with the number of sinusoids being equal to the size of the frequency grid. Clearly, this is not possible without knowing the amplitudes of all the sinusoids, and, as a result, the estimates are formed using an iterative scheme. Following [3], the data covariance matrices are estimated by iteratively estimating the complex amplitudes, $\alpha_i(\omega_k)$, of these sinusoidals, as well as the resulting covariance matrices, $\mathbf{R}_N^{(i)}$, for $i = 1$ or 2, for the respective signals, using (see also [6, 31])

$$\alpha_i(\omega_k) = \frac{\mathbf{f}_N^H(\omega_k)[\mathbf{R}_N^{(i)}]^{-1}\mathbf{x}_N^{(i)}}{\mathbf{f}_N^H(\omega_k)[\mathbf{R}_N^{(i)}]^{-1}\mathbf{f}_N(\omega_k)} \tag{12}$$

$$\mathbf{R}_N^{(i)} = \sum_{k=0}^{K-1} |\alpha_{(i)}(\omega_k)|^2 \mathbf{f}_N(\omega_k)\mathbf{f}_N^H(\omega_k) \tag{13}$$

until practical convergence, for $k = 0, 1, \ldots, K-1$, where the number of grid points $K > N$, with $\mathbf{R}_N^{(i)}$ initialized to the identity matrix $\mathbf{I}_N$. Upon convergence, an estimate of the cross-covariance matrix is then formed as

$$\mathbf{R}_N^{(12)} = \sum_{k=0}^{K-1} \alpha_1^*(\omega_k)\alpha_2(\omega_k)\mathbf{f}_N(\omega_k)\mathbf{f}_N^H(\omega_k) \tag{14}$$

The IAA-based MSC estimator $\gamma_{x_1x_2}^{2,IAA}$ is subsequently estimated as [3]

$$\gamma_{x_1x_2}^{2,IAA}(\omega) = \frac{\left|\mathbf{f}_N^H(\omega)\mathbf{P}_N^{(12)}\mathbf{f}_N(\omega)\right|^2}{\prod_{i=1}^2 \mathbf{f}_N^H(\omega)[\mathbf{R}_N^{(i)}]^{-1}\mathbf{f}_N(\omega)} \tag{15}$$

where

$$\mathbf{P}_N \triangleq [\mathbf{R}_N^{(1)}]^{-1}\mathbf{R}_N^{(12)}[\mathbf{R}_N^{(2)}]^{-1} \tag{16}$$

Direct, brute force, computation of the resulting IAA-based MSC estimates requires a significant amount of computations. Assuming a uniformly spaced frequency grid with K grid points, the cost of forming $\gamma_{x_1x_2}^{2,IAA}(\omega)$ is approximately

$$\mathcal{C}^{IAA} = \mathcal{O}\left(m_{IAA}\left(2N^3 + 4N^2K\right)\right) + \mathcal{O}\left(2N^2K\right) \tag{17}$$

operations, where the parameter m_{IAA} that appears in (17) denotes the number of the IAA iterations.

2.2. A novel SLIM-based MSC estimator

We now proceed to introduce a novel SLIM-based MSC estimator. Similar to the IAA-based approach, the SLIM formulation may be used to form the estimates of the complex amplitudes, $\alpha_i(\omega_k)$, for $i = 1, 2$, of these spectral lines, adopting an l_q-norm based regularized minimization method for sparse signal recovery and noise power estimation, with $0 \leq q \leq 1$, and where a cyclic approach is applied iteratively for the estimation of the data covariance matrices and the complex amplitudes as [7]

$$\alpha_i(\omega_k) = \frac{1}{|\alpha_i(\omega_k)|^{(2-q)}} \mathbf{f}_N^H(\omega_k)[\mathbf{R}_N^{(i)}]^{-1}\mathbf{x}_N^{(i)} \tag{18}$$

$$\mathbf{R}_N^{(i)} = \sum_{k=0}^{K-1} |\alpha_i(\omega_k)|^2 \mathbf{f}_N(\omega_k)\mathbf{f}_N^H(\omega_k) + \eta_i \mathbf{I} \tag{19}$$

$$\eta_i = \frac{1}{N}\left|\mathbf{x}_N^{(i)} - \sum_{k=0}^{K-1} \alpha_i(\omega_k)\mathbf{f}_N(\omega_k)\right|^2 \tag{20}$$

until practical convergence, for $k = 0, 1, \ldots, K-1$, with $\alpha_i(\omega_k)$ initialized to the (zero-padded) Discrete Fourier Transform (DFT) transform of $\mathbf{x}_N$ and η_i initialized to a positive value, usually given values proportional to the signal variances, $\sigma^2_{x^{(i)}}$. Upon convergence, an estimate of the cross-covariance matrix is then formed as in (14), and, finally, the SLIM-based MSC estimate, $\gamma^{2,SLIM}_{x_1x_2}(\omega)$, is computed reminiscent to (15). Assuming a uniformly spaced frequency grid with K grid points, the direct brute force complexity of the SLIM-based MSC estimator requires approximately

$$\mathcal{C}^{SLIM} = \mathcal{O}\left(m_{SLIM}\left(2N^3 + 2N^2K\right)\right) + \mathcal{O}\left(4N^2K\right) \tag{21}$$

operations, with m_{SLIM} denoting the number of the SLIM iterations.

3. MSC estimation using trigonometric polynomials

To reduce the computational complexity of forming the discussed IAA- and SLIM-based MSC estimators, one may exploit the matrix structure to reduce the amount of necessary computations significantly. To do so, note that (10) may be reformulated as

$$\gamma^2_{y_1y_2}(\omega) = \frac{|\varphi_{12}(\omega)|^2}{\varphi_1(\omega)\varphi_2(\omega)} \tag{22}$$

where the trigonometric polynomials $\varphi_i(\omega)$ and $\varphi_{12}(\omega)$ are

$$\varphi_i(\omega) \triangleq \mathbf{f}_N^H(\omega)[\mathbf{R}_N^{(i)}]^{-1}\mathbf{f}_N(\omega) \tag{23}$$

$$\varphi_{12}(\omega) \triangleq \mathbf{f}_N^H(\omega)\mathbf{P}_N\mathbf{f}_N(\omega) \tag{24}$$

for $i = 1$ or 2. As shown in [12, 14, 25], the trigonometric polynomials involved in forming (23) can be efficiently computed by taking into account the displacement representation of the pertinent matrices $\mathbf{R}_N^{(1)}$ and $\mathbf{R}_N^{(2)}$, both of which enjoy a Toeplitz structure. To obtain (22) given these results, it thus only remains to find a computationally efficient way of forming (24), which is directly associated with the structure of the matrix $\mathbf{P}_N$ involved, formed by (16) as the product of Toeplitz like matrices, since $[\mathbf{R}_N^{(1)}]^{-1}$ and $[\mathbf{R}_N^{(2)}]^{-1}$ are Toepliz-like matrices and $\mathbf{R}_N^{(12)}$ is a non-symmetric Toeplitz matrix. For the readers benefit and for further use, we briefly present the basics from the displacement representation theory of matrices (see also [32–34]). Consider a matrix $\mathbf{Q}_N \in \mathcal{C}^{N\times N}$, and define the lower shifting matrix

$$\mathbf{Z}_N = \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{N-1} & \mathbf{0} \end{bmatrix} \tag{25}$$

Clearly, $(\mathbf{Z}_N)^N = \mathbf{0}$. Then, the displacement of $\mathbf{Q}_N$ with respect to $\mathbf{Z}_N$ and $\mathbf{Z}_N^T$ is defined as

$$\nabla_{Z_N,Z_N^T}\mathbf{Q}_N \triangleq \mathbf{Q}_N - \mathbf{Z}_N\mathbf{Q}_N\mathbf{Z}_N^T \tag{26}$$

Suppose that there exist integers ρ and $\sigma_i \in \{-1, 1\}$, $i = 1, 2 \ldots \rho$, such that (see also [32–34])

$$\nabla_{Z_N,Z_N^T}\mathbf{Q}_N = \sum_{i=1}^{\rho} \sigma_i \mathbf{t}_N^i \mathbf{s}_N^{iH} = \mathbf{X}_{N,\rho}^t \mathbf{\Sigma}_\rho \mathbf{X}_{N,\rho}^{sH} \tag{27}$$

where

$$\mathbf{X}_{N,\rho}^t = \begin{bmatrix} \mathbf{t}_N^1 & \ldots & \mathbf{t}_N^\rho \end{bmatrix} \tag{28}$$

$$\mathbf{X}_{N,\rho}^s = \begin{bmatrix} \mathbf{s}_N^1 & \ldots & \mathbf{s}_N^\rho \end{bmatrix} \tag{29}$$

$$\mathbf{\Sigma}_\rho = diag\,(\sigma_1 \ldots \sigma_\rho) \tag{30}$$

where $diag(\mathbf{a})$ denotes the diagonal matrix formed with the vector $\mathbf{a}$ along its diagonal, and with $\mathbf{t}_N^i$ and $\mathbf{s}_N^i$ being the so-called generator vectors. Then, the GS factorization of $\mathbf{Q}_N$ may be expressed as

$$\mathbf{Q}_N = \sum_{i=1}^{\rho} \sigma_i \mathcal{L}(\mathbf{t}_N^i)\mathcal{L}^H(\mathbf{s}_N^i) \tag{31}$$

where $\mathcal{L}(\mathbf{A})$ denotes a Krylov matrix of the form

$$\mathcal{L}(\mathbf{A}) = \left[\begin{array}{cccc} \mathbf{A} & Z_N\mathbf{A} & Z_N^2\mathbf{A} & \ldots & Z_N^{N-1}\mathbf{A} \end{array} \right] \tag{32}$$

The displacement rank of the representation equals the rank of the associated displacement matrix, $\nabla_{Z_N, Z_N^T}\mathbf{Q}_N$, whereas the integer ρ may be larger than or equal to the corresponding displacement rank. In summary, the triplet

$$\left(\mathbf{X}_{N,\rho}^t, \mathbf{X}_{N,\rho}^s, \mathbf{\Sigma}_\rho\right) \tag{33}$$

is called the displacement representation of $\mathbf{Q}_N$ with respect to $\mathbf{Z}_N$ and $\mathbf{Z}_N^T$. Given the displacement representation (33), the coefficients of the trigonometric polynomial associated to $\mathbf{Q}_N$

$$\varphi(\omega) \triangleq \mathbf{f}_N^H(\omega)\mathbf{Q}_N\mathbf{f}_N(\omega) = \sum_{\kappa=-N+1}^{N-1} c_\kappa e^{\jmath\kappa\omega} \tag{34}$$

can efficiently be computed as it is detailed in [25] without the need of forming the matrix $\mathbf{Q}_N$ explicitly, using triangular Toeplitz matrix products of the form

$$\left[\begin{array}{c} c_{N-1} \\ \vdots \\ c_0 \end{array} \right] = \sum_{i=1}^{\rho} \sigma_i \boldsymbol{\mathcal{L}}(\mathbf{t}_N^i)\boldsymbol{\mathcal{D}}_N \left(\mathbf{s}_N^i\right)^* \tag{35}$$

$$\left[\begin{array}{c} c_{-N+1} \\ \vdots \\ c_0 \end{array} \right] = \sum_{i=1}^{\rho} \sigma_i \boldsymbol{\mathcal{L}} \left(\left(\mathbf{s}_N^i\right)^*\right) \boldsymbol{\mathcal{D}}_N \mathbf{t}_N^i \tag{36}$$

and where $\boldsymbol{\mathcal{D}}_N$ is a $N \times N$ anti-diagonal matrix of the form

$$\boldsymbol{\mathcal{D}}_N = \left[\begin{array}{cccc} 0 & \ldots & 0 & 1 \\ 0 & \ldots & 2 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ N & 0 & \ldots & 0 \end{array} \right]. \tag{37}$$

It is worth noting that (34) can be subsequently evaluated on the unit circle using the Fast Fourier Transform (FFT) [35].

4. Implementation of the IAA- and SLIM-based MSC estimator

As shown in [12–14], $\mathbf{R}_N^{(i)}$ may be extracted from a circulant matrix of higher dimensions, such that

$$\mathbf{C}_K^{(i)} \triangleq \mathbf{W}_K^H \text{diag}\left\{|\alpha_i(\omega_0)|^2, \ldots, |\alpha_i(\omega_{K-1})|^2\right\} \mathbf{W}_K = \begin{bmatrix} \mathbf{R}_N^{(i)} & \times \\ \times & \times \end{bmatrix}$$

where $\mathbf{W}_K$ is the DFT matrix of size $K \times K$, and the symbol $\times$ denotes unspecified terms of no relevance. Thus, the first column of $\mathbf{R}_N^{(i)}$, denoted by $\mathbf{r}_N^{(i)}$, is obtained by the partition $\mathbf{c}_K^{(i)} = [\mathbf{r}_N^{(i)T} \ \times]^T$, where $\mathbf{c}_K^{(i)}$ is the first column of $\mathbf{C}_K^{(i)}$, and can be computed using the Inverse DFT (IDFT) as $\mathbf{c}_K^{(i)} = \mathbf{W}_K^H \boldsymbol{\alpha}_i$, where

$$\boldsymbol{\alpha}_i \triangleq \begin{bmatrix} |\alpha_i(\omega_0)|^2 & \ldots & |\alpha_i(\omega_{K-1})|^2 \end{bmatrix}^T \tag{38}$$

The cross-covariance matrix $\mathbf{R}_N^{(12)}$ can be treated similarly, and its first column and row are extracted from a circulant matrix whose first column is computed as $\mathbf{c}_K^{(12)} = \mathbf{W}_K^H \boldsymbol{\alpha}_{12}$, where

$$\boldsymbol{\alpha}_{12} \triangleq \begin{bmatrix} \alpha_1^*(\omega_0)\alpha_2(\omega_0) & \ldots & \alpha_1^*(\omega_{K-1})\alpha_2(\omega_{K-1}) \end{bmatrix}^T \tag{39}$$

Introduce the partitioning

$$\mathbf{R}_N^{(i)} = \begin{bmatrix} r_0^{(i)} & \mathbf{r}_{N-1}^{f(i)H} \\ \mathbf{r}_{N-1}^{f(i)} & \mathbf{R}_{N-1}^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{N-1}^{(i)} & \mathbf{J}_{N-1}\mathbf{r}_{N-1}^{f(i)*} \\ \mathbf{r}_{N-1}^{f(i)T}\mathbf{J}_{N-1} & r_0^{(i)} \end{bmatrix} \tag{40}$$

where $\mathbf{J}_{N-1}$ is the exchange matrix and define

$$\hat{\mathbf{a}}_N^{(i)} = \begin{bmatrix} 1 \\ -[\mathbf{R}_{N-1}^{(i)}]^{-1}\mathbf{r}_{N-1}^{f(i)} \end{bmatrix} / \sqrt{\alpha_N^{f(i)}} \tag{41}$$

$$\alpha_N^{f(i)} = r_0^{(i)} - \mathbf{r}_{N-1}^{f(i)H}[\mathbf{R}_{N-1}^{(i)}]^{-1}\mathbf{r}_{N-1}^{(i)} \tag{42}$$

Then, a displacement representation of $[\mathbf{R}_N^{(1)}]^{-1}$ and $[\mathbf{R}_N^{(2)}]^{-1}$ with respect to $\mathbf{Z}_N$ and $\mathbf{Z}_N^T$ may be formed as $\left(\mathbf{T}_{N,2}^{(i)}, \mathbf{T}_{N,2}^{(i)}, \boldsymbol{\Sigma}_2^R\right)$, where

$$\mathbf{T}_{N,2}^{(i)} = \begin{bmatrix} \mathbf{t}_N^{(i),1} & \mathbf{t}_N^{(i),2} \end{bmatrix} \tag{43}$$

$$\boldsymbol{\Sigma}_2^R = diag(\sigma_1^R, \sigma_2^R) \tag{44}$$

$$\mathbf{t}_N^{(i),1} = \hat{\mathbf{a}}_N^{(i)}, \quad \sigma_1^R = 1 \tag{45}$$

$$\mathbf{t}_N^{(i),2} = \mathbf{Z}_N\mathbf{J}_N\left(\mathbf{t}_N^{(i),2}\right)^*, \quad \sigma_2^R = -1 \tag{46}$$

yielding a displacement rank of $\rho^R = 2$ (see also [29, 32]). The resulting fast implementation for the computation of (12) and (13) may then be computed using the methods presented in [12, 14, 25], at a cost of $m_{IAA}\left[N^2 + 12\phi(2N) + 3\phi(K)\right]$ operations, where $\phi(N)$ denotes the complexity of forming the FFT of length N. To complete the derivation of the proposed MSC algorithm, it remains to show how the trigonometric polynomial $\varphi_{12}(\omega_k)$, defined in (24), can be computed efficiently. Since $[\mathbf{R}_N^{(i)}]^{-1}$ and $\mathbf{R}_N^{(12)}$ have displacement ranks of two with respect to $\mathbf{Z}_N$ and $\mathbf{Z}_N^T$, the displacement rank of their product $\mathbf{P}_N$, given in (16), has an upper bound of eight [34], although, as is proved in Appendix A, it is actually allowing a lower rank representation, as:

Lemma 1. *A displacement representation of* $\mathbf{P}_N$ *with respect to* $\mathbf{Z}_N$ *and* $\mathbf{Z}_N^T$ *may be formed as* $\left(\boldsymbol{\Upsilon}_{N,4}, \boldsymbol{\Psi}_{N,4}, \Sigma_4^P\right)$*, where*

$$\boldsymbol{\Upsilon}_{N,4} = \left[\begin{array}{ccc} \boldsymbol{v}_N^1 & \ldots & \boldsymbol{v}_N^4 \end{array}\right] \tag{47}$$

$$\boldsymbol{\Psi}_{N,4} = \left[\begin{array}{ccc} \boldsymbol{\psi}_N^1 & \ldots & \boldsymbol{\psi}_N^4 \end{array}\right] \tag{48}$$

$$\boldsymbol{\Sigma}_4^P = diag(\sigma_1^P, \ldots \sigma_4^P) \tag{49}$$

with the auxiliary variables formed in Table 1, and

$$\begin{aligned} \boldsymbol{v}_N^1 &= \hat{\mathbf{a}}_N^{(1)}, \quad \boldsymbol{\psi}_N^1 = \mathbf{d}_N^{(12)}, \quad \sigma_1^P = 1 \\ \boldsymbol{v}_N^2 &= \mathbf{d}_N^{(2)}, \quad \boldsymbol{\psi}_N^2 = \hat{\mathbf{a}}_N^{(2)}, \quad \sigma_2^P = 1 \\ \boldsymbol{v}_N^3 &= \mathbf{Z}_N \mathbf{J}_N \hat{\mathbf{a}}_N^{(1)*}, \quad \boldsymbol{\psi}_N^3 = \mathbf{Z}_N \mathbf{c}_N^{(12)}, \quad \sigma_3^p = -1 \\ \boldsymbol{v}_N^4 &= \mathbf{Z}_N \mathbf{c}_N^{(2)}, \quad \boldsymbol{\psi}_N^4 = \mathbf{Z}_N \mathbf{J}_N \hat{\mathbf{a}}_N^{(2)*}, \quad \sigma_4^P = -1 \end{aligned}$$

The displacement rank of the representation is thus $\rho^P = 4$. □

The proof can be found in the appendix. Moreover, given the displacement representation of $[\mathbf{R}_N^{(i)}]^{-1}$, a displacement representation of

$$\left[\begin{array}{cc} [\mathbf{R}_{N-1}^{(i)}]^{-1} & 0 \\ 0^T & 0 \end{array}\right] \tag{50}$$

for $i = 1, 2$, with respect to $\mathbf{Z}_N$ and $\mathbf{Z}_N^T$, may be formed as $\left(\tilde{\mathbf{T}}_{N,2}^{(i)}, \tilde{\mathbf{T}}_{N,2}^{(i)}, \boldsymbol{\Sigma}_2^R\right)$, where

$$\tilde{\mathbf{T}}_{N,2}^{(i)} = \left[\begin{array}{cc} \mathbf{t}_N^{(i),1} & \mathbf{J}_N \left(\mathbf{t}_N^{(i),1}\right)^* \end{array}\right] \tag{51}$$

Given this GS factorization, the coefficients of the associated polynomial in (24) may be efficiently computed using the FFT as detailed in [25], at a complexity of no more than $40\phi(2N)$ operations. The computational cost for computing the variables that appear in Table 1 using these techniques is no more than $56\phi(2N)$ operations, implying that the total computational cost for the fast implementation of the IAA-based MSC algorithm, hereafter termed FIAA-MSC, is no more than

$$\mathcal{C}^{FIAA} = 2m_{IAA}\left[N^2 + 12\phi(2N) + 3\phi(K)\right] + 64\phi(2N) + 2\phi(K) \tag{52}$$

where the first term corresponds to the complexity of the two IAA algorithms required for the computation of the autocorrelation sequences as well as the associated trigonometric polynomials, whereas the last term corresponds to the additional computations required for the computation of the cross correlation between the two signals and the associated trigonometric polynomial. The fast IAA-based MSC estimation is tabulated in Table 3. The structure of the SLIM-based MSC estimator is very similar to that of the IAA-based estimator. The (auto) covariances of the signals involved are then estimated using the SLIM iterations in (18)-(20), while the cross-covariance, and the MSC estimate are computed using (14) and (15), which are common to both approaches. The SLIM iterations in (18)-(20) can be efficiently implemented using a similar approach as described above for the implementation of the IAA iterations since these share a similar structure, noting that the resulting scheme is somewhat simpler than that of the IAA-based estimate, yielding a computational complexity of

$$\mathcal{C}^{FSLIM} = 2m_{SLIM}\left[N^2 + 2\phi(K)\right] + 74\phi(2N) + 4\phi(K)$$

operations. As shown in Figure 1, the proposed fast IAA- and SLIM-based MSC implementations are about three orders of magnitude faster than their brute force counterparts.

5. The case of missing samples

The MSC estimators presented so far have been designed for evenly sampled data. However, in a wide range of applications the measured data could be unevenly sampled, or might suffer from lost samples. We refer to this as the missing samples MSC estimation case. Several algorithms for spectral estimation in the arbitrarily missing data case have been reported, such as the missing data APES (MAPES) algorithm [36] and the missing data IAA

(MIAA) algorithm [31], with the former suffering from high computational complexity and from performance degradation when the percentage of missing samples increases, whereas the latter has been found to provide accurate estimates even when only a few data snapshots are available, allowing furthermore for fast implementations as has been recently shown in [14] and [15] for the 1-D and 2-D cases, respectively. In [15], in addition to the 2-D MIAA, the missing data SLIM (MSLIM) algorithm is presented and its fast implementation is derived. The performance of MAPES, MIAA, and MSLIM for spectral estimation of 1-D and 2-D data sets in the case of arbitrarily missing data has been investigated in [15, 31] by extensive simulations, illustrating that (i) MIAA and MSLIM both perform much better than MAPES, and (ii) the computational complexity of MIAA and MSLIM are several orders of magnitude lower than that of the MAPES algorithm. For this reason, we will here focus on the IAA- and SLIM-based MSC estimators and their efficient implementation, extending these algorithms to data sets that contain missing samples, where the locations of the missing samples are arbitrarily selected, but known. Consider the vectors of available (or given) data, defined as

$$\mathbf{x}_{N_g^i}^{(i)} = \mathbf{S}_{N_g^i,N}\mathbf{x}_N^{(i)}, \quad i = 1, 2 \tag{53}$$

where $\mathbf{S}_{N_g^i N}$ is a $N_g^i \times N$ selection matrix, with zeros and ones indicating the presence or absence of a sample, respectively, and $\mathbf{S}_{N_g^i,N}\mathbf{S}_{N_g^i,N}^T = \mathbf{I}_{N_g^i}$, where $\mathbf{I}_{N_g^i}$ is the $N_g^i \times N_g^i$ identity matrix, and where, in general, $N_g^1 \neq N_g^2$, allowing thus for a different missing data pattern for each signal $x_1(n)$ and $x_2(n)$ (see also, e.g., [14, 15, 31, 36]). Here, $N_g^i \leq N$ denotes the number of available data samples for each signal. The corresponding frequency vector can be expressed as

$$\mathbf{f}_{N_g^i}^{(i)}(\omega) = \mathbf{S}_{N_g^i,N}\mathbf{f}_N(\omega) \tag{54}$$

The MIAA-MSC algorithm is then formed by iterating [3]

$$\alpha_i(\omega_k) = \frac{\mathbf{f}_{N_g^i}^{(i)H}(\omega_k)[\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{x}_{N_g^i}^{(i)}}{\mathbf{f}_{N_g^i}^{(i)H}(\omega_k)[\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{f}_{N_g^i}^{(i)}(\omega_k)} \triangleq \frac{\psi_i(\omega)}{\phi_i(\omega)} \tag{55}$$

$$\mathbf{R}_{N_g^i}^{(i)} = \sum_{k=0}^{K-1} |\alpha_i(\omega_k)|^2 \mathbf{f}_{N_g^i}^{(i)}(\omega_k)\mathbf{f}_{N_g^i}^{(i)H}(\omega_k) \tag{56}$$

until convergence, followed by the MSC estimator as

$$\gamma_{x_1x_2}^{2,MIAA}(\omega) = \frac{\left|\tilde{\mathcal{G}}_{\omega}^{(1)H}\mathbf{R}_{N_g^1N_g^2}^{12}\tilde{\mathcal{G}}_{\omega}^{(2)}\right|^2}{\prod_{i=1}^{2}\mathbf{f}_{N_g^i}^{(i)H}(\omega)[\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{f}_{N_g^i}^{(i)}(\omega)} \tag{57}$$

where $\tilde{\mathcal{G}}_{\omega}^{(i)} = [\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{f}_{N_g^i}(\omega)$, with

$$\mathbf{R}_{N_g^1N_g^2}^{(12)} = \sum_{k=0}^{K-1}\alpha_1^*(\omega_k)\alpha_2(\omega_k)\mathbf{f}_{N_g^1}^{(1)}(\omega_k)\mathbf{f}_{N_g^2}^{(2)H}(\omega_k) \tag{58}$$

being a $N_g^1 \times N_g^2$ Toeplitz matrix. The complexity of a brute-force implementation of the MIAA-based MSC estimator is

$$\mathcal{C}^{MIAA} = \mathcal{O}\left(m_{MIAA}\sum_{i=1}^{2}\left((N_g^i)^3 + 2(N_g^i)^2K\right)\right) + \mathcal{O}(N_g^1N_g^2K) \tag{59}$$

operations, with m_{MIAA} denoting the required number of MIAA iterations. Computational reduction of the MIAA-based MSC estimator can be achieved by first applying a fast scheme for the computation of the auto- and cross-correlation sequences related to the signals $x_1(n)$ and $x_2(n)$ from the missing data and by subsequently bypassing (57) and using (15) directly for the estimation of the relevant MSC. Using the fact that

$$\mathbf{R}_{N_g^i}^{(i)} = \mathbf{S}_{N_g^i,N}\mathbf{R}_N^{(i)}\mathbf{S}_{N_g^i,N}^T \tag{60}$$

and noting that the factors of (55) can be expressed as

$$\begin{aligned}\psi_i(\omega_k) &= \mathbf{f}_N^H(\omega_k)\left(\mathbf{S}_{N_g^i,N}^T[\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{x}_{N_g^i}\right)\\ \phi_i(\omega_k) &= \mathbf{f}_N^H(\omega_k)\left(\mathbf{S}_{N_g^i,N}^T[\mathbf{R}_{N_g^i}^{(i)}]^{-1}\mathbf{S}_{N_g^i,N}\right)\mathbf{f}_N(\omega_k)\end{aligned}$$

an algorithm for the fast estimation of $\alpha_i(\omega)$ in (55), as well as the auto-correlation sequences of the signals $x_1(n)$ and $x_2(n)$, has been recently presented in [14], and combined with a fast implementation of (58) using the Toeplitz into circulant matrix embedding technique discussed in the previous section, results in a fast scheme of the MIAA-based MSC estimation, whose complexity is given by

$$\begin{aligned}\mathcal{C}^{FMIAA} &= \mathcal{O}\left(m_{MIAA}\left((N_g^1)^3 + (N_g^2)^3 + 6K\phi(K)\right)\right) +\\ &\quad \mathcal{O}\left(2N^2 + 20\phi(2N) + 4\phi(K)\right)\end{aligned} \tag{61}$$

The missing data SLIM-based MSC estimator is formed along the lines of the MIAA-based MSC, using (18)-(20) reformulated to the missing data formulation, for the estimation of the auto-correlation sequence of $x_1(n)$ and $x_2(n)$, requiring

$$\mathcal{C}^{MSLIM} = \mathcal{O}\left(m_{MSLIM}\sum_{i=1}^{2}\left((N_g^i)^3 + (N_g^i)^2 K\right)\right) + \mathcal{O}(N_g^1 N_g^2 K) \tag{62}$$

and

$$\begin{aligned}\mathcal{C}^{FMSLIM} &= \mathcal{O}\left(m_{MSLIM}\left((N_g^1)^3 + (N_g^2)^3 + 4K\phi(K)\right)\right) + \\ &\quad \mathcal{O}\left(2N^2 + 20\phi(2N) + 4\phi(K)\right)\end{aligned} \tag{63}$$

operations, for the direct and for the fast implementation, respectively, with m_{MSLIM} denoting the required number of MSLIM iterations.

6. Extensions to 2-D data sets

We proceed to examine 2-D extensions of the IAA- and the SLIM-based MSC estimator, $\gamma^2_{x_1x_2}(\omega_{k_1}, \omega_{k_2})$, of two stationary complex valued 2-D signals, $x_1(n_1, n_2)$ and $x_2(n_1, n_2)$, for $n_1 = 0, 1, \ldots, N_1 - 1$ and $n_2 = 0, 1, \ldots, N_2 - 1$. Let the 2-D data matrices

$$\begin{aligned}\mathbf{X}^{(i)}_{N_1,N_2} &= \left[\begin{array}{ccc} \mathbf{x}^{(i)}_{N_1}(0) & \ldots & \mathbf{x}^{(i)}_{N_1}(N_2-1)\end{array}\right] \\ \mathbf{x}^{(i)}_{N_1}(n_2) &= \left[\begin{array}{ccc} x_i(0,n_2) & \ldots & x_i(N_1-1,n_2)\end{array}\right]^T\end{aligned}$$

where $i = 1, 2$ and $n_2 = 0, 1, \ldots, N_2 - 1$, be organized in a column-wise form as $\mathbf{x}^{(i)}_{N_1N_2} = \mathrm{vec}(\mathbf{X}^{(i)}_{N_1,N_2})$, where $\mathrm{vec}(\cdot)$ denotes column-wise vectorization. Furthermore, define the 2-D frequency vector $\mathbf{f}_{N_1N_2}(\omega_1, \omega_2) \triangleq \mathbf{f}_{N_2}(\omega_2) \otimes \mathbf{f}_{N_1}(\omega_1)$, where $\otimes$ denotes the Kronecker product. The 2-D IAA-based MSC estimator is then given by

$$\gamma^{2,IAA}_{x_1x_2}(\omega_1, \omega_2) = \frac{\left|\mathbf{f}^H_{N_1N_2}(\omega_1, \omega_2)\mathbf{P}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_1, \omega_2)\right|^2}{\prod_{i=1}^{2}\mathbf{f}^H_{N_1N_2}(\omega_1, \omega_2)[\mathbf{R}^{(i)}_{N_1N_2}]^{-1}\mathbf{f}_{N_1N_2}(\omega_1, \omega_2)} \tag{64}$$

with

$$\mathbf{P}_{N_1N_2} \triangleq [\mathbf{R}^{(1)}_{N_1N_2}]^{-1}\mathbf{R}^{(12)}_{N_1N_2}[\mathbf{R}^{(2)}_{N_1N_2}]^{-1} \tag{65}$$

where the 2-D auto- and cross-covariance matrices are estimated using the 2-D IAA algorithm, iterating

$$\alpha_{i,k_1,k_2} = \frac{\mathbf{f}_{N_1N_2}^{(k_1,k_2)H}[\mathbf{R}_{N_1N_2}^{(i)}]^{-1}\mathbf{x}_{N_1N_2}^{(i)}}{\mathbf{f}_{N_1N_2}^{(k_1,k_2)H}[\mathbf{R}_{N_1N_2}^{(i)}]^{-1}\mathbf{f}_{N_1N_2}^{(k_1,k_2)}} \tag{66}$$

$$\mathbf{R}_{N_1N_2}^{(i)} = \sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1} |\alpha_{i,k_1,k_2}|^2 \mathbf{f}_{N_1N_2}^{(k_1,k_2)}\mathbf{f}_{N_1N_2}^{(k_1,k_2)H} \tag{67}$$

until practical convergence, where $\mathbf{f}_{N_1N_2}^{k_1,k_2} \triangleq \mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2})$ and $\alpha_{i,k_1,k_2} \triangleq \alpha_i(\omega_{k_1},\omega_{k_2})$, whereas upon convergence, an estimate of the cross-covariance matrix is formed as

$$\mathbf{R}_{N_1N2}^{(12)} = \sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1} \alpha_{1,k_1,k_2}^{*}\alpha_{2,k_1,k_2}\mathbf{f}_{N_1N_2}^{(k_1,k_2)}\mathbf{f}_{N_1N_2}^{(k_1,k_2)H}. \tag{68}$$

The 2-D SLIM-based MSC estimator is formed by, instead of (66) and (67), the estimation of the amplitude spectra and the auto-correlation matrices of the 2-D signals is performed by the 2-D-SLIM algorithm, by iteratively estimating

$$\alpha_{i,k_1,k_2} = \frac{1}{|\alpha_{i,k_1,k_2}|^{(2-q)}}\mathbf{f}_{N_1N_2}^{(k_1,k_2)H}[\mathbf{R}_{N_1N_2}^{(i)}]^{-1}\mathbf{x}_{N_1N_2}^{(i)} \tag{69}$$

$$\mathbf{R}_{N_1N_2}^{(i)} = \sum_{k_1=0}^{K-1}\sum_{k_2=0}^{K_2-1} |\alpha_{i,k_1,k_2}|^2 \mathbf{f}_{N_1N_2}^{(k_1,k_2)}\mathbf{f}_{N_1N_2}^{(k_1,k_2)H} + \eta_i\mathbf{I} \tag{70}$$

$$\eta_i = \frac{1}{N_1N_2}\left|\mathbf{x}_{N_1N_2}^{(i)} - \sum_{k_1=0}^{K-1}\sum_{k_2=0}^{K_2-1}\alpha_{i,k_1,k_2}\mathbf{f}_{N_1N_2}^{(k_1,k_2)}\right|^2 \tag{71}$$

until convergence. Fast implementations of both the 2-D IAA and the 2-D SLIM algorithms have been recently proposed in [12, 14, 15] where, thanks to the Toeplitz-block-Toeplitz (TBT) structure of the auto-correlation matrices (67) and (70), the 2-D auto-correlation sequences are estimating using TBT to 2-D circulant matrix embedding, their inverses are expressed using a suitable 2-D GS representation whose generators are computed using the Levinson-Whittle-Wiggins-Robinson (LWWR) algorithm [28, 37, 38], which are subsequently used for the computation of the coefficients of the pertinent trigonometric polynomials using [25], resulting in a very efficient implementation with significantly reduced computational complexity as compared to the brute force approach. Furthermore, (68) is a TBT matrix whose block cross-correlation sequence may be computed efficiently using TBT to 2-D circular matrix embedding and the 2-D FFT. Thus, it remains to show how (64) may be

implemented efficiently. Reminiscent to the 1-D case, one may express (64) in terms of the 2-D trigonometric polynomials as

$$\gamma_{x_1x_2}^{2,IAA}(\omega_1,\omega_2) = \frac{|\varphi_{12}(\omega_1,\omega_2)|^2}{\varphi_1(\omega_1,\omega_2)\varphi_2(\omega_1,\omega_2)} \tag{72}$$

where

$$\varphi_i(\omega_1,\omega_2) \triangleq \mathbf{f}_{N_1N_2}^H(\omega_1,\omega_2)[\mathbf{R}_{N_1N_2}^{(i)}]^{-1}\mathbf{f}_{N_1N_2}(\omega_1,\omega_2) \tag{73}$$

$$\varphi_{12}(\omega_1,\omega_2) \triangleq \mathbf{f}_{N_1N_2}^H(\omega_1,\omega_2)\mathbf{P}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_1,\omega_2) \tag{74}$$

for $i = 1, 2$. The polynomials $\varphi_1(\omega_1,\omega_2)$ and $\varphi_2(\omega_1,\omega_2)$ are already available as these are inherently involved in the iterative scheme of the 2-D IAA algorithm in (66) and (67), whereas for the 2-D SLIM-based MSC estimator, they are efficiently estimated from the 2-D GS representation of $[\mathbf{R}_{N_1N_2}^{(i)}]^{-1}$. Thus, it remains to show how $\varphi_{12}(\omega_1,\omega_2)$ may be efficiently computed, which, similarly to the 1-D case presented above, reduces to the efficient estimation of the 2-D GS representation of the matrix $\mathbf{P}_{N_1N_2}$ as defined by (65). Consider the partitions of the $(N_1N_2)\times(N_1N_2)$ TBT matrix in (67) as

$$\mathbf{R}_{N_1N_2}^{(i)} = \begin{bmatrix} \mathbf{R}_{N_1(N_2-1)}^{(i)} & \boldsymbol{\mathcal{R}}^{b(i)} \\ \boldsymbol{\mathcal{R}}^{b(i)H} & \mathbf{R}^{0(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{0(i)} & \boldsymbol{\mathcal{R}}^{f(i)H} \\ \boldsymbol{\mathcal{R}}^{f(i)} & \mathbf{R}_{N_1(N_2-1)} \end{bmatrix} \tag{75}$$

where $\mathbf{R}_{N_1(N_2-1)}^{(i)}$ is a TBT matrix of dimensions $(N_1(N_2-1))\times(N_1(N_2-1))$, which, using the matrix inversion lemma for partitioned matrices, may be expressed as

$$[\mathbf{R}_{N_1N_2}^{(i)}]^{-1} = \begin{bmatrix} [\mathbf{R}_{N_1(N_2-1)}^{(i)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix} + \bar{\boldsymbol{\mathcal{B}}}_{N_1N_2}^{(i)}\bar{\boldsymbol{\mathcal{B}}}_{N_1N_2}^H = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & [\mathbf{R}_{N_1N_2}^{(i)}]^{-1} \end{bmatrix} + \bar{\boldsymbol{\mathcal{A}}}_{N_1N_2}\bar{\boldsymbol{\mathcal{A}}}_{N_1N_2}^H$$

where $\bar{\boldsymbol{\mathcal{B}}}_{N_1N_2}^{(i)}$ and $\bar{\boldsymbol{\mathcal{A}}}_{N_1N_2}^{(i)}$ are block matrices of dimensions $N_1N_2\times N_1$ defined by

$$\bar{\boldsymbol{\mathcal{B}}}_{N1N_2}^{(i)} = \begin{bmatrix} \mathbf{I}_{N_1} \\ -[\mathbf{R}_{N_1(N_2-1)}^{(i)}]^{-1}\boldsymbol{\mathcal{R}}^{b(i)} \end{bmatrix} [\mathbf{A}_{N_1}^{b(i)}]^{-1/2}, \tag{76}$$

$$\bar{\boldsymbol{\mathcal{A}}}_{N1N_2} = \begin{bmatrix} -[\mathbf{R}_{N_1(N_2-1)}^{(i)}]^{-1}\boldsymbol{\mathcal{R}}^{f(i)} \\ \mathbf{I}_{N_1} \end{bmatrix} [\mathbf{A}_{N_1}^{(i)}]^{-1/2} \tag{77}$$

$$\mathbf{A}_{N_1}^{b(i)} = \mathbf{R}^{0(i)} - \boldsymbol{\mathcal{R}}^{b(i)H}[\mathbf{R}_{N_1(N_2-1)}^{(i)}]^{-1}\boldsymbol{\mathcal{R}}^{b(i)} \tag{78}$$

$$\mathbf{A}_{N_1}^{f(i)} = \mathbf{R}^{0(i)} - \boldsymbol{\mathcal{R}}^{f(i)H}[\mathbf{R}_{N_1(N_2-1)}^{(i)}]^{-1}\boldsymbol{\mathcal{R}}^{f(i)} \tag{79}$$

with $\mathbf{A}^{-1/2}$ denoting the Cholesky factor of $\mathbf{A}^{-1}$. Using (76) and (77), and working along the lines of Lemma 1, a displacement representation of (65) with respect to the block shifting operators $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}^T_{N_1N_2}$, where $\mathbf{Z}_{N_1N_2} = \mathbf{Z}_{N_2} \otimes \mathbf{I}_{N_1}$, takes the form $\left(\boldsymbol{\mathcal{U}}_{N_1N_2,4N_2}, \boldsymbol{\mathcal{V}}_{N_1N_2,4N_2}, \Sigma^P_{4N_2}\right)$, where

$$\boldsymbol{\mathcal{U}}_{N_1N_2,4N_2} = \left[\begin{array}{ccc} \boldsymbol{\mathcal{U}}^1_{N_1N_2} & \ldots & \boldsymbol{\mathcal{U}}^4_{N_1N_2} \end{array} \right] \tag{80}$$

$$\boldsymbol{\mathcal{V}}_{N_1N_2,4} = \left[\begin{array}{ccc} \boldsymbol{\mathcal{V}}^1_{N_1N_2} & \ldots & \boldsymbol{\mathcal{V}}^4_{N_1N_2} \end{array} \right] \tag{81}$$

$$\Sigma^P_{4N_2} = diag(\sigma^P_1 \mathbf{I}_{N_2}, \ldots \sigma^P_4 \mathbf{I}_{N_2}) \tag{82}$$

with $\sigma^P_1 = \sigma^P_2 = 1$, $\sigma^P_3 = \sigma^P_4 = -1$, and

$$\begin{aligned}
\boldsymbol{\mathcal{U}}^1_{N_1N_2} &= \bar{\boldsymbol{\mathcal{A}}}^1_{N_1N_2}, \quad \boldsymbol{\mathcal{V}}^1_{N_1N_2} = \boldsymbol{\mathcal{T}}_{N_1N_2} \\
\boldsymbol{\mathcal{U}}^2_{N_1N_2} &= \boldsymbol{\mathcal{D}}^2_{N_1N_2}, \quad \boldsymbol{\mathcal{V}}^2_{N_1N_2} = \bar{\boldsymbol{\mathcal{A}}}^2_{N_1N_2} \\
\boldsymbol{\mathcal{U}}^3_{N_1N_2} &= \mathbf{Z}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}^1_{N_1N_2}, \quad \boldsymbol{\mathcal{V}}^3_{N_1N_2,N_2} = \mathbf{Z}_{N_1N_2}\boldsymbol{\mathcal{S}}_{N_1N_2} \\
\boldsymbol{\mathcal{U}}^4_{N_1N_2} &= \mathbf{Z}_{N_1N_2}\boldsymbol{\mathcal{C}}^2_{N_1N_2}, \quad \boldsymbol{\mathcal{V}}^4_{N_1N_2} = \mathbf{Z}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}^2_{N_1N_2}
\end{aligned}$$

where the required auxiliary variables are tabulated in Table 2. The displacement rank of the representation is $\rho^P = 4N_2$, and the computational cost of the resulting 2-D IAA- and SLIM-based MSC algorithms reduces to

$$\begin{aligned}
\mathcal{C}^{FIAA_2} &= \mathcal{O}\left(m_{FIAA_2}\left(3N_2^2N_1^3 + 10N_1\phi(2N_1,2N_2) + 6\phi(K_1,K_2)\right)\right) + \\
&\quad 19N_1^3N_2 + 18N_1\phi(2N_1,2N_2) + \phi(K_1,K_2) \\
\mathcal{C}^{FSLIM_2} &= \mathcal{O}\left(m_{FSLIM_2}\left(3N_2^2N_1^3 + 4\phi(K_1,K_2)\right)\right) + \\
&\quad 19N_1^3N_2 + 28N_1\phi(2N_1,2N_2) + 3\phi(K_1,K_2)
\end{aligned}$$

where m_{FIAA_2} and m_{FSLIM_2} denote the number of required iterations. As it is illustrated in Fig. 2, this is a substantial improvement as compared to the brute force approach, which requires

$$\begin{aligned}
\mathcal{C}^{IAA_2} &= \mathcal{O}\left(m_{IAA_2}\left(2N_1^3N_2^3 + 6N_1^2N_2^2K_1K_2\right)\right) + 2N_1^2N_2^2K_1K_2 \\
\mathcal{C}^{SLIM_2} &= \mathcal{O}\left(m_{SLIM_2}\left(2N_1^3N_2^3 + 4N_1^2N_2^2K_1K_2\right)\right) + 4N_1^2N_2^2K_1K_2
\end{aligned}$$

where m_{IAA_2} and m_{SLIM_2} denote the corresponding number of required iterations.

7. Numerical examples

The performance of the proposed MSC estimation algorithms are illustrated by means of computer simulations. Initially, we examine the 1-D complete data case, and consider $N = 200$ samples of two signals, $x_1(n)$ and $x_2(n)$, which are both a mixture of sinusoidal signals corrupted by additive noise

$$x_i(n) = \sum_{\ell=1}^{7} r_\ell^i e^{\jmath 2\pi f_\ell^i n} + w_i(n), \quad i = 1, 2 \tag{83}$$

where r_ℓ^i are complex amplitudes of unit magnitude and uniformly distributed phases, and with $w_i(n)$, for $i = 1, 2$, denoting two independent circularly symmetric zero-mean Gaussian random processes with unit variance. Here, the signals' frequencies are selected as $f^1 = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.31 & 0.6 & 0.61 & 0.8 \end{bmatrix}$ and $f^2 = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0 & 0.6 & 0.61 & 0 \end{bmatrix}$. The MSC is evaluated over $K = 1000$ uniformly distributed frequency grid points. The Capon- and APES- based MSC estimates are illustrated in Fig. 3(a) and (b) with filter lengths of $M = 40$, where the MSC is successfully resolved by either method, noting that the high noise floor (erroneous peaks) can be reduced by decreasing the value of M at the expense of lower spectral resolution. The IAA-based MSC with $m_{IAA} = 10$ IAA iterations is shown in Fig. 3(c), where the MSC produced by the IAA-MSC method can be seen to be characterized by an extremely high noise floor, which can be somewhat reduced by applying data segmentation and averaging as proposed in [3]. Fig. 3(d) illustrates this for data segments of size $N_s = 120$, although, as is clear from the figure, without significant reducing the noise floor. In an attempt to reduce the variance of the IAA-based MSC estimates, a reduced size MSC estimator may be formed by using a smaller fraction $N_R < N$ of the full sized cross-correlation sequence computed by the IAA algorithm [39], which is subsequently utilized for the computation of the MSC. Thus, instead of using the full sized approach in (16) and (24), a lower order estimator is adopted using

$$\hat{\varphi}_{12}(\omega) \triangleq \mathbf{f}_{N_R}^H(\omega)\mathbf{P}_{N_R}\mathbf{f}_{N_R}(\omega) \tag{84}$$

$$\mathbf{P}_{N_R} \triangleq [\mathbf{R}_{N_R}^{(1)}]^{-1}\mathbf{R}_{N_R}^{(12)}[\mathbf{R}_{N_R}^{(2)}]^{-1} \tag{85}$$

with $N_R \leq N$, while keeping $\varphi_1(\omega)$ and $\varphi_2(\omega)$ as originally defined in (23), resulting in

$$\gamma_{x_1x_2}^{2,IAA-I}(\omega) = \frac{|\hat{\varphi}_{12}(\omega)|^2}{\varphi_1(\omega)\varphi_2(\omega)} \tag{86}$$

which is hereafter termed the IAA-I-based MSC algorithm. Due to the order recursive structure of the Levinson-Durbin algorithm, lower order GS factorizations are produced at no extra cost. Moreover, the computation of the displacement of $\mathbf{P}_{N_R}$ required for the efficient computation of (84) is in this case lower than that of original full order approach. It is worth noticing that the auto- and cross-correlation sequences of the input signals are still estimated using the full order IAA algorithms. Using the proposed order reduction technique, the variance of the MSC is drastically reduced as it is shown in Fig. 3(e), where $N_s = 50$ is used. Finally, the performance of the proposed SLIM-based MSC algorithm is shown in Fig. 3(f), noting that in this approach the noise floor using the full sized cross-correlation sequence is insignificant and that there is thus no need to consider a similar order reduction modification as just proposed for the IAA-based MSC case. Figures 4 and 5 further illustrates this for varying SNR levels. We proceed to examine the performance of the proposed IAA- and SLIM-based MSC algorithms in the missing data case, by randomly (with a uniform distribution) omitting 30% and 70% samples, where, as shown in Fig. 6, both methods succeed in obtaining MSC estimates similar to those obtained in the full data case. Finally, the performance of the proposed 2-D MSC estimation methods is illustrated in Fig. 7, where two 2-D mixtures of sinusoidal signals corrupted by additive noise are considered, formed as

$$x_i(n_1, n_2) = \sum_{\ell=1}^{4} r_\ell^i e^{\jmath 2\pi \left(f_{\ell,1}^i n_1 + f_{\ell,2}^i n_2\right)} + w_i(n_1, n_2), \quad i = 1, 2$$

where r_ℓ^i are the complex amplitudes of unit magnitude and uniformly distributed phases, and the 2-D frequencies of the signals are

$$\begin{aligned} f^1 &= \Big\{(0.1, 0.1) \quad (0.2, 0.2) \quad (0.3, 0.3) \quad (0.4, 0.4)\Big\} \\ f^2 &= \Big\{(0.1, 0.4) \quad (0.2, 0.3) \quad (0.3, 0.3) \quad (0.4, 0.1)\Big\} \end{aligned}$$

respectively, and where $w_i(n_1, n_2)$, for $i = 1, 2$, are two independent 2-D circularly symmetric zero-mean Gaussian random processes with unit variance. As for the 1-D case, the 2-D IAA-based MSC is estimated by using a reduced size cross-correlation estimate in place of the full size counterpart in (74) as

$$\hat{\varphi}_{12}(\omega_1, \omega_2) \triangleq \mathbf{f}_{N_{R_1} N_{R_2}}^H(\omega_1, \omega_2) \mathbf{P}_{N_{R_1} N_{R_2}} \mathbf{f}_{N_{R_1} N_{R_2}}(\omega_1, \omega_2)$$

where $N_{R_1} \leq N_1$, $N_{R_2} \leq N_2$, and

$$\mathbf{P}_{N_{R_1}N_{R_2}} \triangleq [\mathbf{R}^{(1)}_{N_{R_1}N_{R_2}}]^{-1}\mathbf{R}^{(12)}_{N_{R_1}N_{R_2}}[\mathbf{R}^{(2)}_{N_{R_1}N_{R_2}}]^{-1} \tag{87}$$

while keeping the full size auto-correlation estimates as originally defined in (73), resulting in MSC estimates with significantly reduced variance as compared to the original scheme. Reminiscent to the 1-D case, we term this the 2-D IAA-I based MSC algorithm. The 2-D MSC estimates obtained by the 2-D Capon, 2-D APES, 2-D IAA-I and the 2-D SLIM based MSC algorithms are presented in Fig. 7.

8. Conclusions

This works examines the performance of, as well as introduce computationally efficient implementations for, the IAA-based MSC estimators, as well as for the here introduced SLIM-based MSC estimator. We furthermore introduce efficient implementations of the estimators for the case of missing samples as well as for 2-D data sets, in both cases offering substantial computational savings as compared to a direct evaluation. Reduced form IAA-based formulation for both 1-D and 2-D data sets are also introduced, offering notably better estimates as compared to the full sized cross-correlation versions. Numerical simulations illustrate both the achievable reduction in computational complexity and typical performance of the estimators for some narrowband data sets.

A. Proof of Lemma 1

Application of the matrix inversion lemma for partitioned matices on (40) yields

$$[\mathbf{R}^{(i)}_N]^{-1} = \begin{bmatrix} [\mathbf{R}^{(i)}_{N-1}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{\mathbf{J}_N\hat{\mathbf{a}}^{(i)*}_N\hat{\mathbf{a}}^{(i)T}_N\mathbf{J}_N}{\alpha^{f(i)}_N} = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & [\mathbf{R}^{(i)}_{N-1}]^{-1} \end{bmatrix} + \frac{\hat{\mathbf{a}}^{(i)}_N\hat{\mathbf{a}}^{(i)H}_N}{\alpha^{f(i)}_N}$$

which combined with (16) yields the upper and lower partitions

$$\mathbf{P}_N = \begin{bmatrix} \mathbf{P}_{N-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \mathbf{J}_N\hat{\mathbf{a}}^{(1)*}_N\mathbf{c}^{(12)H}_N + \mathbf{c}^{(2)}_N\hat{\mathbf{a}}^{(2)T}_N\mathbf{J}_N = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{P}_{N-1} \end{bmatrix} + \hat{\mathbf{a}}^{(1)}_N\mathbf{c}^{(12)H}_N + \mathbf{d}^{(2)}_N\hat{\mathbf{a}}^{(2)H}_N$$

allowing for the displacement of $\mathbf{P}_N$ to be formed as

$$\mathbf{P}_N - \mathbf{Z}_N\mathbf{P}_N\mathbf{Z}^T_N = \hat{\mathbf{a}}^{(1)}_N\mathbf{d}^{(12)H}_N + \mathbf{d}^{(2)}_N\hat{\mathbf{a}}^{(2)H}_N - \mathbf{Z}_N\mathbf{J}_N\hat{\mathbf{a}}^{(1)*}_N\mathbf{c}^{(12)H}_N\mathbf{Z}^T_N - \mathbf{Z}_N\mathbf{c}^{(2)}_N\hat{\mathbf{a}}^{(2)T}_N\mathbf{J}_N\mathbf{Z}^T_N$$

References

[1] J. Benesty, J. Chen, Y. Huang, A Generalized MVDR Spectrum, IEEE Signal Processing Letters 12 (2005) 827–830.

[2] A. Jakobsson, S. R. Alty, J. Benesty, Estimating and Time-Updating the 2-D Coherence Spectrum, IEEE Transactions on Signal Processing 55 (2007) 2350–2354.

[3] N. R. Butt, A. Jakobsson, Coherence Spectrum Estimation From Nonuniformly Sampled Sequences, IEEE Signal Processing Letters 17 (2010) 339–342.

[4] M. Zhou, C. Zheng, X. Li, On the relationship of non-parametric methods for coherence function estimation, Signal Processing 88 (2008) 2863–2867.

[5] C. Zheng, Y. Zhou, X. Li, Generalised framework for nonparametric coherence function estimation, Electronics Letters 46 (2010) 450–452.

[6] T. Yardibi, J. Li, P. Stoica, M. Xue, A. B. Baggeroer, Source Localization and Sensing: A Nonparametric Iterative Approach Based on Weighted Least Squares, IEEE Transactions on Aerospace and Electronic Systems 46 (2010) 425–443.

[7] X. Tan, W. Roberts, J. Li, P. Stoica, Sparse Learning via Iterative Minimization With Application to MIMO Radar Imaging, IEEE Transactions on Signal Processing 59 (2011) 1088–1101.

[8] I. F. Gorodnitsky, B. D. Rao, Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-weighted Minimum Norm Algorithm, IEEE Transactions on Signal Processing 45 (1997) 600–616.

[9] S. I. Adalbjörnsson, A. Jakobsson, Sparse Estimation of Spectroscopic Signals, in: 19th European Signal Processing Conference, EUSIPCO 2011, Barcelona, Spain.

[10] W. Roberts, P. Stoica, J. Li, T. Yardibi, F. A. Sadjadi, Iterative Adaptive Approaches to MIMO Radar Imaging, IEEE Journal of Selected Topics in Signal Processing 4 (2010) 5–20.

[11] E. Gudmundson, A. Jakobsson, J. A. Jensen, P. Stoica, Blood Velocity Estimation Using Ultrasound and Spectral Iterative Adaptive Approaches, Signal Process. 91 (2011) 1275–1283.

[12] M. Xue, L. Xu, J. Li, IAA Spectral Estimation: Fast Implementation using the Gohberg-Semencul Factorization, IEEE Transactions on Signal Processing 59 (2011) 3251 – 3261.

[13] G.-O. Glentis, A. Jakobsson, Time-Recursive IAA Spectral Estimation, IEEE Signal Processing Letters 18 (2011) 111–114.

[14] G.-O. Glentis, A. Jakobsson, Efficient Implementation of Iterative Adaptive Approach Spectral Estimation Techniques, IEEE Transactions on Signal Processing 59 (2011) 4154–4167.

[15] D. Vu, L. Xu, M. Xue, J. Li, Nonparametric Missing Sample Spectral Analysis and Its Applications to Interrupted SAR, IEEE Journal of Selected Topics in Signal Processing 6 (2012) 1–14.

[16] G.-O. Glentis, A. Jakobsson, Superfast Approximative Implementation of the IAA Spectral Estimate, IEEE Transactions on Signal Processing 60 (2012) 472–478.

[17] G. O. Glentis, K. Zhao, A. Jakobsson, J. Li, Non-Parametric High-Resolution SAR Imaging, IEEE Transactions on Signal Processing (????). To appear.

[18] K. Angelopoulos, G. O. Glentis, A. Jakobsson, Computationally Efficient Capon- and APES-based Coherence Spectrum Estimation, IEEE Transactions on Signal Processing (????). To appear.

[19] J. Li, P. Stoica, An Adaptive Filtering Approach to Spectral Estimation and SAR Imaging, IEEE Transactions on Signal Processing 44 (1996) 1469–1484.

[20] Z. S. Liu, H. Li, J. Li, Efficient implementation of Capon and APES for spectral estimation, IEEE Transactions on Aerospace and Electronic Systems 34 (1998) 1314–1319.

[21] R. Wu, Z.-S. Liu, J. Li, Time-varying complex spectral analysis via recursive APES, IEE Proc. Radar, Sonar and Navigation 145 (1998) 354–360.

[22] A. Jakobsson, S. L. Marple, Jr., P. Stoica, Two-Dimensional Capon Spectrum Analysis, IEEE Transactions on Signal Processing 48 (2000) 2651–2661.

[23] E. G. Larsson, P. Stoica, Fast Implementation of Two-Dimensional APES and Capon Spectral Estimators, Multidimensional Systems and Signal Processing 13 (2002) 35–54.

[24] S. R. Alty, A. Jakobsson, E. G. Larsson, Efficient Time-Recursive Implementation of Matched Filterbank Spectral Estimators, IEEE Transactions on Circuits and Systems—Part I: Regular Papers 52 (2005) 516–521.

[25] G.-O. Glentis, A Fast Algorithm for APES and Capon Spectral Estimation, IEEE Transactions on Signal Processing 56 (2008) 4207–4220.

[26] G. O. Glentis, Efficient Algorithms for Adaptive Capon and APES Spectral Estimation, IEEE Transactions on Signal Processing 58 (2010) 84–96.

[27] S. M. Kay, Modern Spectral Estimation: Theory and Application, Prentice-Hall, Englewood Cliffs, N.J., 1988.

[28] J. S. L. Marple, Digital spectral analysis with applications, Prentice Hall, Englewood Cliffs, NJ, 1987.

[29] P. Stoica, R. Moses, Spectral Analysis of Signals, Prentice Hall, Upper Saddle River, N.J., 2005.

[30] P. Stoica, A. Jakobsson, J. Li, Matched-Filterbank Interpretation of Some Spectral Estimators, Signal Processing 66 (1998) 45–59.

[31] P. Stoica, J. Li, J. Ling, Missing Data Recovery via a Nonparametric Iterative Adaptive Approach, IEEE Signal Processing Letters 16 (2009) 241–244.

[32] T. Kailath, A. H. Sayed, Displacement Structure: Theory and Applications, SIAM Review 37 (1995) 297–386.

[33] I. Gohberg, V. Olshevksy, Complexity of multiplication with vectors for structured matrices, Linear Algebra Appl. 202 (1994) 163–192.

[34] D. Wood, Product rules for the displacement of near-Toeplitz matrices, Linear Algebra Appl. 188/189 (1993) 641–663.

[35] H. Li, P. Stoica, J. Li, Capon Estimation of Covariance Sequences, Circuits, Systems, and Signal Processing 17 (1998) 29–49.

[36] Y. Wang, J. Li, P. Stoica, Spectral Analysis of Signals - The Missing Data Case, Morgan & Claypool, 2005.

[37] R. A. Wiggins, E. A. Robinson, Recursive Solution of the Multichannel Filtering Problem, J. Geophys. Res. 70 (1965) 1885–1891.

[38] N. Kalouptsidis, G. Carayannis, D. Manolakis, Fast Algorithms for Block Toeplitz Matrices with Toeplitz Entries, Signal Process. 6 (1984) 77–81.

[39] K. Angelopoulos, G. O. Glentis, A. Jakobsson, Efficient Implementation of the IAA-based Magnitude Squared Coherence Estimator, in: International Conference on Digital Signal Processing, Corfu.

Table 1: Auxiliary variables required for the displacement representation of $\mathbf{P}_N$.

$$
\begin{aligned}
\beta_N &= \hat{\mathbf{a}}_N^{(1)T}\mathbf{J}_N\mathbf{R}_N^{(12)}\mathbf{J}_N\hat{\mathbf{a}}_N^{(2)*} \\
\mathbf{c}_N^{(1)} &= \begin{bmatrix} [\mathbf{R}_{N-1}^{(2)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{R}_N^{(12)H}\mathbf{J}_N\hat{\mathbf{a}}_N^{(1)*} \\
\mathbf{c}_N^{(2)} &= \begin{bmatrix} [\mathbf{R}_{N-1}^{(1)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{R}_N^{(12)}\mathbf{J}_N\hat{\mathbf{a}}_N^{(2)*} \\
\mathbf{c}_N^{(12)} &= \mathbf{c}_N^{(1)} + \mathbf{J}_N\hat{\mathbf{a}}_N^{(2)*}\beta_N^* \\
\delta_N &= \hat{\mathbf{a}}_N^{(1)H}\mathbf{R}_N^{(12)}\hat{\mathbf{a}}_N^{(2)} \\
\mathbf{d}_N^{(1)} &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & [\mathbf{R}_{N-1}^{(2)}]^{-1} \end{bmatrix} \mathbf{R}_N^{(12)H}\hat{\mathbf{a}}_N^{(1)} \\
\mathbf{d}_N^{(2)} &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & [\mathbf{R}_{N-1}^{(1)}]^{-1} \end{bmatrix} \mathbf{R}_N^{(12)}\hat{\mathbf{a}}_N^{(2)} \\
\mathbf{d}_N^{(12)} &= \mathbf{d}_N^{(1)} + \hat{\mathbf{a}}_N^{(2)}\delta_N^*
\end{aligned}
$$

Table 2: Auxiliary variables required for the displacement representation of $\mathbf{P}_{N_1N_2}$.

$$
\begin{aligned}
\mathbf{B}_{N_1} &= \bar{\boldsymbol{\mathcal{B}}}^{(1)H}_{N_1N_2}\mathbf{R}^{(12)}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}^{(2)}_{N_1N_2} \\
\boldsymbol{\mathcal{C}}^{(1)}_{N_1N_2} &= \begin{bmatrix} [\mathbf{R}^{(2)}_{N_1(N_2-1)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{R}^{(12)H}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}^{(2)}_{N_1N_2} \\
\boldsymbol{\mathcal{C}}^{(2)}_{N_1N_2} &= \begin{bmatrix} [\mathbf{R}^{(1)}_{N_1(N_2-1)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{R}^{(12)}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}^{(2)}_{N_1N_2} \\
\boldsymbol{\mathcal{S}}_{N_1N_2} &= \boldsymbol{\mathcal{C}}^{1}_{N_1N_2} + \bar{\boldsymbol{\mathcal{B}}}^{(2)}_{N_1N_2}\mathbf{B}_{N_1} \\
\boldsymbol{\Delta}_{N_1N_2} &= \bar{\boldsymbol{\mathcal{A}}}^{(1)H}_{N_1N_2}\mathbf{R}^{(12)}_{N_1N_2}\bar{\boldsymbol{\mathcal{A}}}^{(2)}_{N_1N_2} \\
\boldsymbol{\mathcal{D}}^{(1)}_{N_1N_2} &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & [\mathbf{R}^{(2)}_{N_1(N_2-1)}]^{-1} \end{bmatrix} \mathbf{R}^{(12)H}_{N_1N_2}\bar{\boldsymbol{\mathcal{A}}}^{(1)}_{N_1N_2} \\
\boldsymbol{\mathcal{D}}^{(2)}_{N_1N_2} &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & [\mathbf{R}^{(1)}_{N_1(N_2-1)}]^{-1} \end{bmatrix} \mathbf{R}^{(12)}_{N_1N_2}\bar{\boldsymbol{\mathcal{A}}}^{(2)}_{N_1N_2} \\
\boldsymbol{\mathcal{T}}_{N_1N_2} &= \boldsymbol{\mathcal{D}}^{(1)}_{N_1N_2} + \bar{\boldsymbol{\mathcal{A}}}^{(2)}_{N_1N_2}\boldsymbol{\Delta}^{*}_{N_1}
\end{aligned}
$$

Table 3: Fast IAA-based MSC estimation. (3.6) and (3.6) are actually implemented using the Levinson-Durbin algorithm. (3.2),(3.3),(3.5), (3.11) and (3.16) are computed using the FFT. (3.9),(3.10),(3.14) and (3.15) are computed using fast Toeplitz vector multiplication.

INITIALIZATION

$$\mathbf{y}_N^{(i)} = \mathbf{x}_N^{(i)}, \quad c_0^{(i)} = N, \quad c_{\kappa\neq 0}^{(i)} = 0 \quad (3.1)$$

FOR $m = 1$ TO m_{IAA}

$$\psi_i(\omega_k) = \mathbf{f}_N^H(\omega_k)\mathbf{y}_N^{(i)} \quad (3.2)$$

$$\varphi_i(\omega_k) = \sum_{l=-N+1}^{N-1} c_l^{(i)} e^{\jmath l\omega_k} \quad (3.3)$$

$$\alpha_i(\omega_k) = \frac{\psi_i(\omega_k)}{\varphi_i(\omega_k)} \quad (3.4)$$

$$\left[r_0^{(i)} \ \mathbf{r}_{N-1}^{(i)T} \times \right]^T = \mathbf{W}_K^H \left[\ |\alpha_i(\omega_0)|^2 \ \ \ldots \ \ |\alpha_i(\omega_{K-1})|^2 \ \right]^T \quad (3.5)$$

$$\hat{\mathbf{a}}_N^{(i)} = \begin{bmatrix} 1 \\ -[\mathbf{R}_{N-1}^{(i)}]^{-1}\mathbf{r}_{N-1}^{f(i)} \end{bmatrix} / \sqrt{\alpha_N^{f(i)}} \quad (3.6)$$

$$\alpha_N^{f(i)} = r_0^{(i)} - \mathbf{r}_{N-1}^{f(i)H}[\mathbf{R}_{N-1}^{(i)}]^{-1}\mathbf{r}_{N-1}^{(i)} \quad (3.7)$$

$$\mathbf{t}_N^{(i),1} = \hat{\mathbf{a}}_N^{(i)}, \quad \mathbf{t}_N^{(i),2} = \mathbf{Z}_N\mathbf{J}_N\left(\mathbf{t}_N^{(i),2}\right)^* \quad (3.8)$$

$$\mathbf{y}_N^{(i)} = \left(\sum_{\ell=1}^{2} \sigma_\ell^R \boldsymbol{\mathcal{L}}(\mathbf{t}_N^{(i),\ell})\boldsymbol{\mathcal{L}}^H(\mathbf{t}_N^{(i),\ell}) \right) \mathbf{x}_N^{(i)} \quad (3.9)$$

$$\left[c_{N-1}^{(i)} \ \ldots c_0^{(i)} \right]^T = \sum_{\ell=1}^{2} \sigma_\ell^R \boldsymbol{\mathcal{L}}(\mathbf{t}_N^{(i),\ell})\boldsymbol{\mathcal{D}}_N\left(\mathbf{t}_N^{(i),\ell}\right)^*, \quad c_{-\kappa}^{(i)} = c_\kappa^{(i)*} \quad (3.10)$$

END

$$\left[\mathbf{r}_N^{(12)T} \times \bar{\mathbf{r}}_{N-1}^{(12)T} \right]^T = \mathbf{W}_K^H \left[\ \alpha_1(\omega_0)\alpha_2^*(\omega_0) \ \ \ldots \ \ \alpha_1(\omega_{K-1})\alpha_2^*(\omega_{K-1}) \ \right]^T \quad (3.11)$$

COMPUTE (3.12) AND (3.13) USING TABLE 1

$$\boldsymbol{\Upsilon}_{N,4} = \left[\ \boldsymbol{v}_N^1 \ \ \ldots \ \ \boldsymbol{v}_N^4 \ \right] \quad (3.12)$$

$$\boldsymbol{\Psi}_{N,4} = \left[\ \boldsymbol{\psi}_N^1 \ \ \ldots \ \ \boldsymbol{\psi}_N^4 \ \right] \quad (3.13)$$

$$\left[c_{N-1}^{(12)} \ \ldots c_0^{(12)} \right]^T = \sum_{\ell=1}^{4} \sigma_\ell^P \boldsymbol{\mathcal{L}}(\boldsymbol{v}_N^\ell)\boldsymbol{\mathcal{D}}_N\left(\boldsymbol{\psi}_N^\ell\right)^* \quad (3.14)$$

$$\left[c_{-N+1}^{(12)} \ \ldots c_0^{(12)} \right]^T = \sum_{\ell=1}^{4} \sigma_\ell^P \boldsymbol{\mathcal{L}}\left(\left(\boldsymbol{\psi}_N^\ell\right)^*\right)\boldsymbol{\mathcal{D}}_N\boldsymbol{v}_N^\ell \quad (3.15)$$

$$\varphi_{12}(\omega_k) = \sum_{l=-N+1}^{N-1} c_l^{(12)} e^{\jmath l\omega_k} \quad (3.16)$$

$$\gamma_{y_1y_2}^2(\omega) = \frac{|\varphi_{12}(\omega)|^2}{\varphi_1(\omega)\varphi_2(\omega)} \quad (3.17)$$

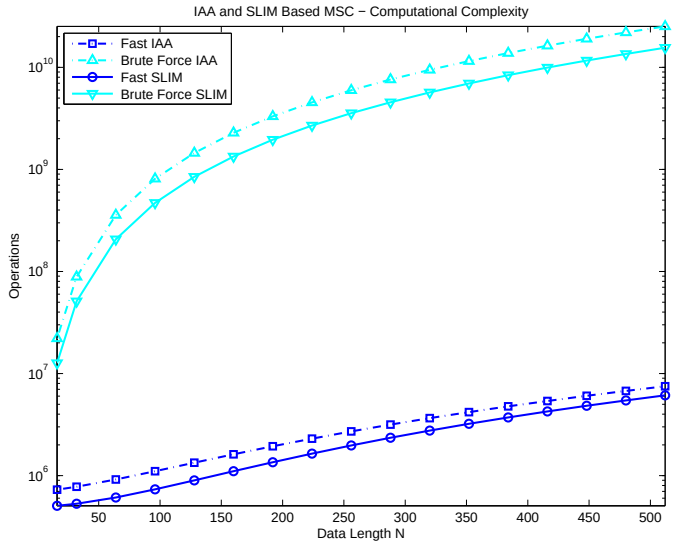


Figure 1: Computational complexity of the IAA- and SLIM-based MSC algorithms using the proposed and the brute force implementations, for different N, where $m = 10$, and with $K = 2048$.

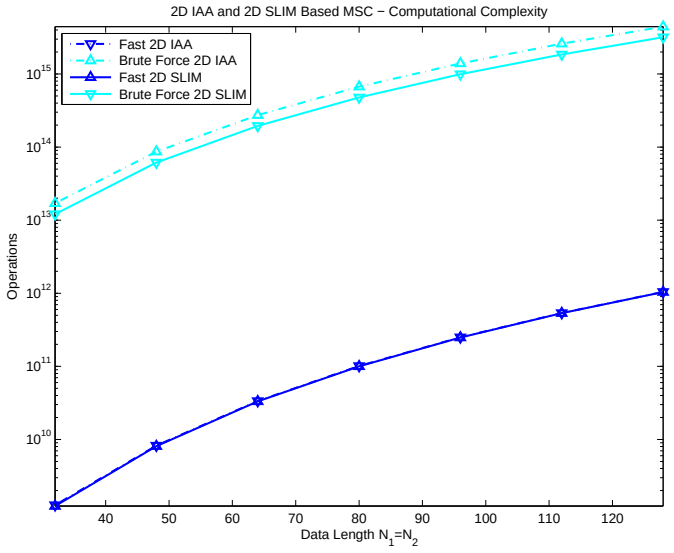
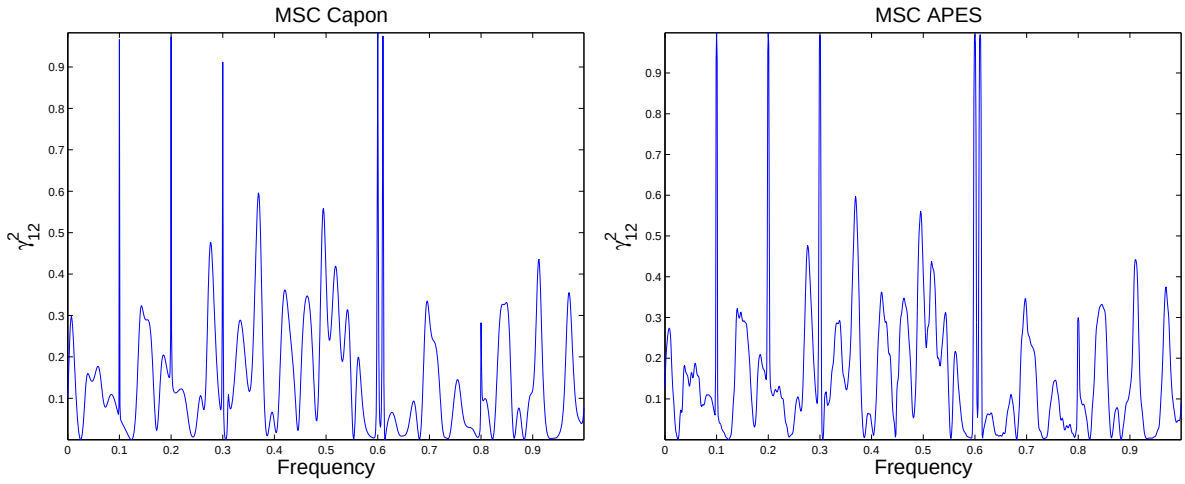
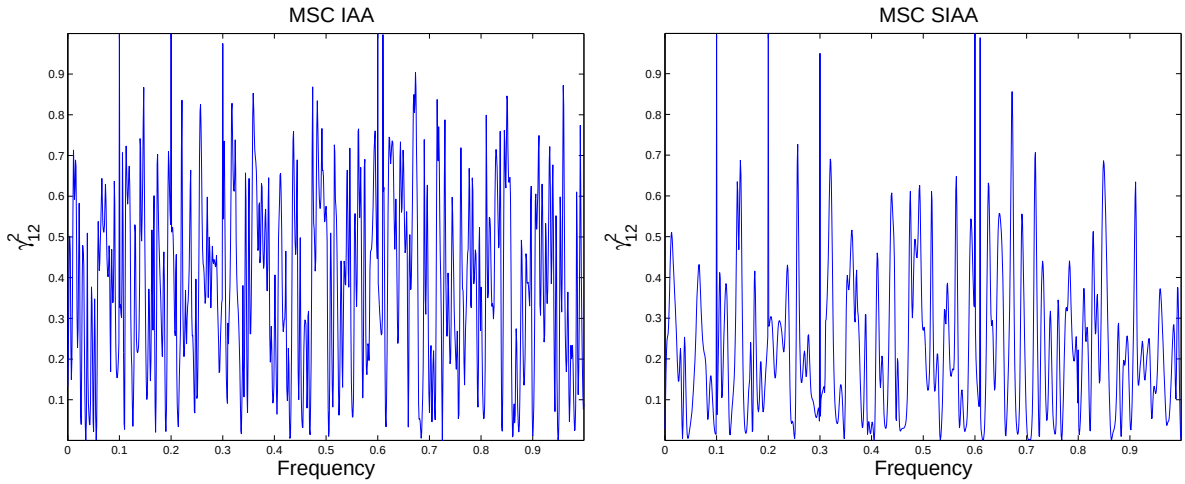


Figure 2: Computational complexity of the IAA- and SLIM-based 2-D MSC algorithms using the proposed and the brute force implementations, for different image dimensions, with $N_1 = N_2$, where $m = 10$, and with $K_1 = K_2 = 512$.



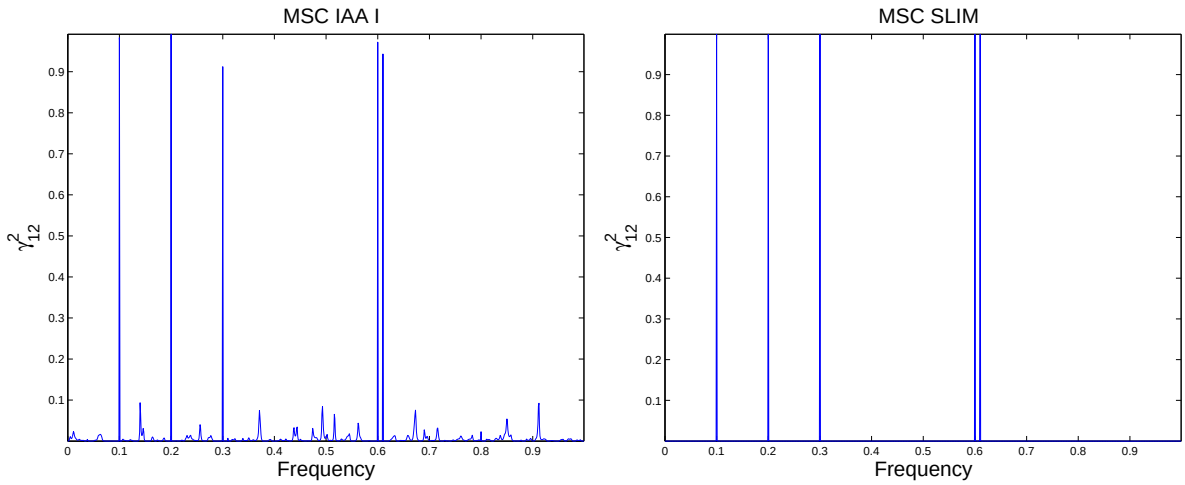
(a)

(b)



(c)

(d)



(e)

(f)

Figure 3: MSC estimation of two cisoidal mixtures using $N = 200$, at SNR=5dB, with $K = 1000$ uniformly spaced frequency points: (a), (b) Capon- and APES-based MSC with filter lengths set equal to $M = 40$, (c) IAA-MSC, (d) Segmented IAA-MSC, with segment lengths $N_S = 120$, (e) IAA-I MSC with $N_R = 50$, and (f) SLIM-MSC.

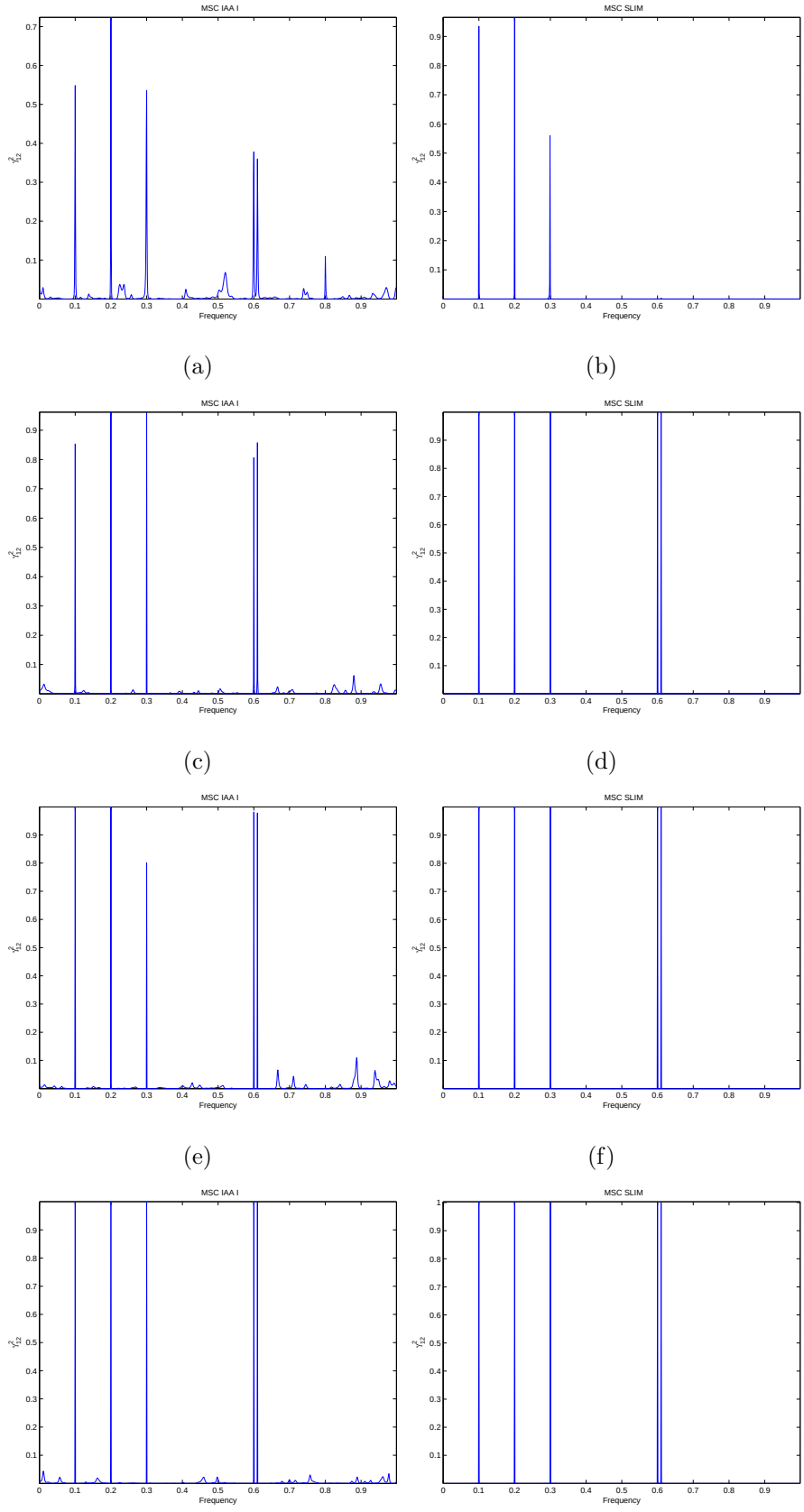


Figure 4: MSC estimation of two cisoidal mixtures using $N = 100$ with $K = 1000$ uniformly spaced frequency points: IAA-I MSC with $N_R = 25$ at SNR varying from 3dB, 10dB, 15dB and 25dB illustrated in figures (a), (c), (e) and (g) respectively, SLIM-MSC illustrated in figures (b), (d), (f) and (h) for the same SNR.

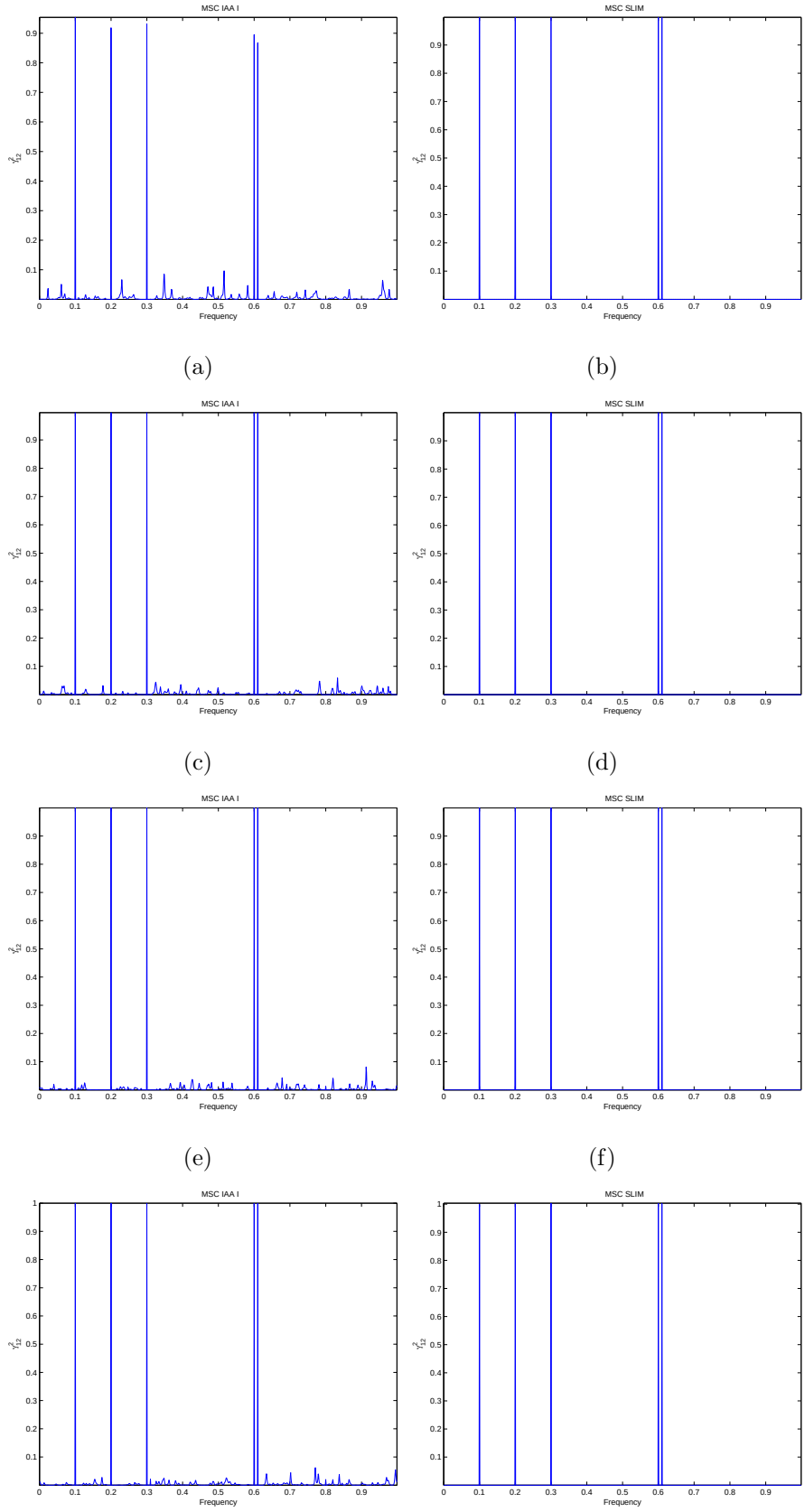
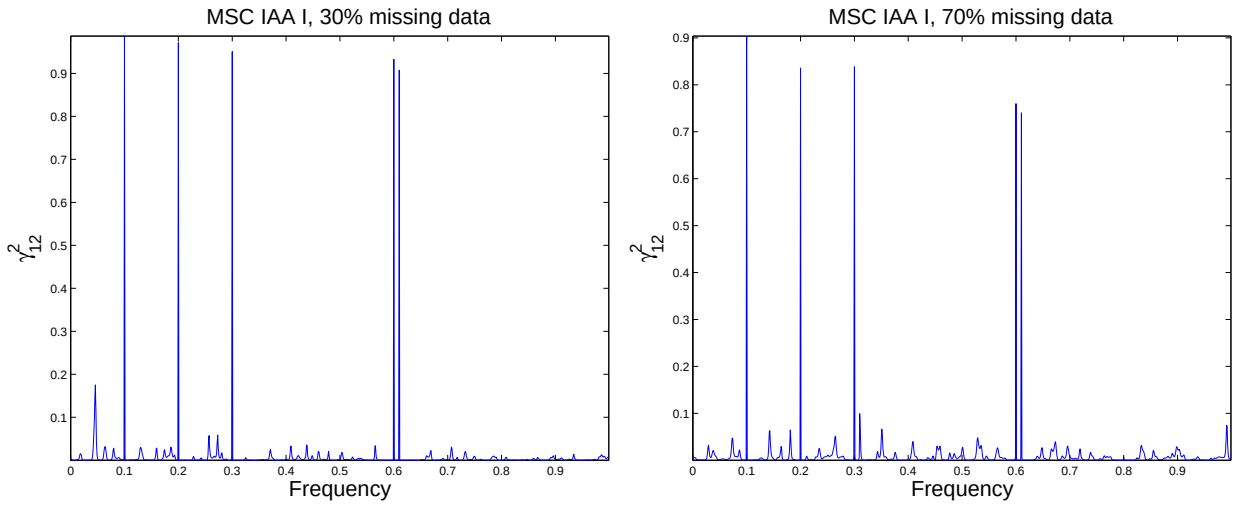
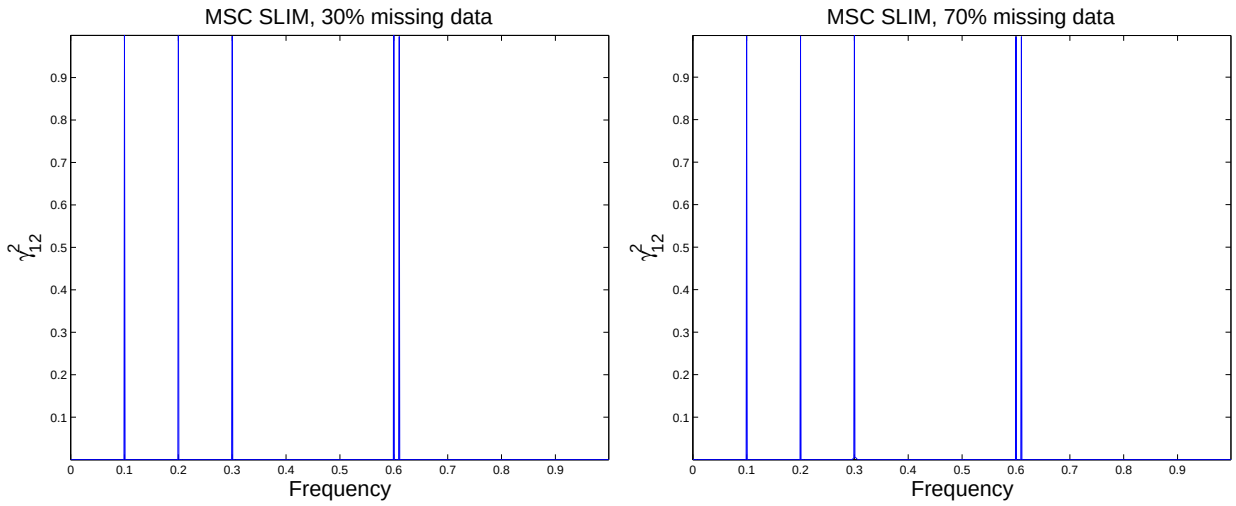


Figure 5: MSC estimation of two cisoidal mixtures using $N = 200$ with $K = 1000$ uniformly spaced frequency points: IAA-I MSC with $N_R = 50$ at SNR varying from 3dB, 10dB, 15dB and 25dB illustrated in figures (a), (c), (e) and (g) respectively, SLIM-MSC illustrated in figures (b), (d), (f) and (h) for the same SNR.

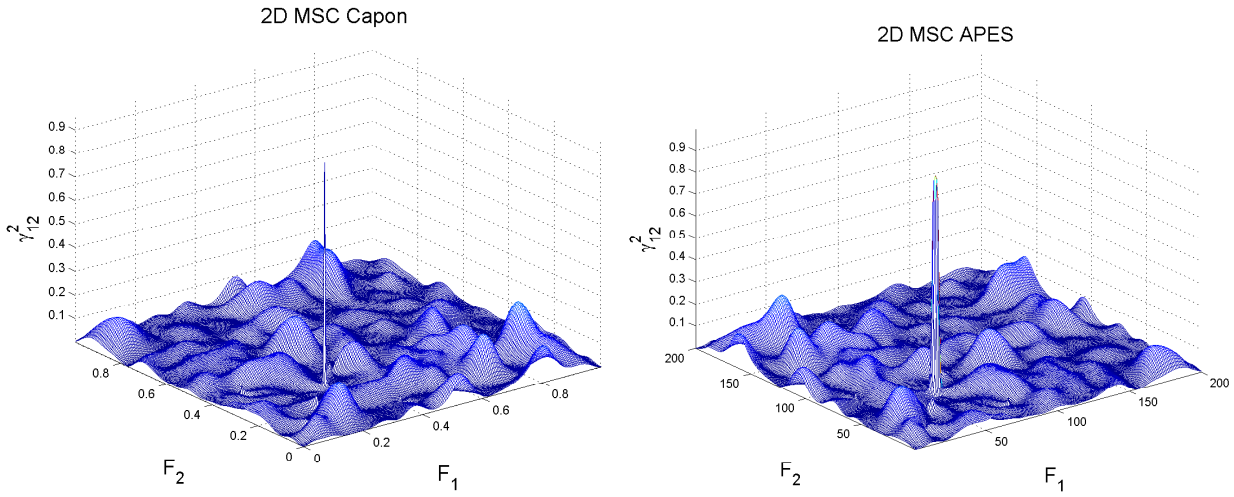


(a) (b)

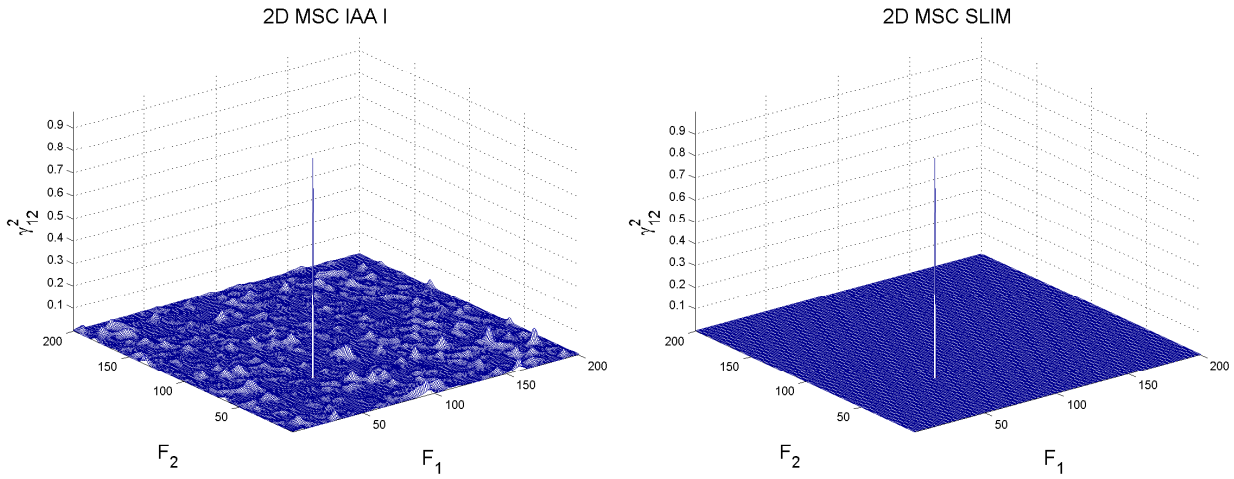


(c) (d)

Figure 6: MSC estimation of two cisoidal mixtures using $N = 200$, at SNR=5dB, with $K = 1000$ uniformly spaced frequency points: (a), (b) IAA-I MSC with $N_R = 50$ where 30% and 70% of data are missing, (c), (d) SLIM MSC with $N_R = 50$ where 30% and 70% of data are missing,



(a)



(b)

Figure 7: MSC estimation of two 2-D cisoidal mixtures using $[N_1 \times N_2] = [32 \times 32]$, at SNR=5dB, with $[K_1 \times K_2] = [200 \times 200]$ uniformly spaced 2-D frequency points: (a), (b) 2-D Capon and 2-D APES based MSC with filter lengths set equal to $[M_1 \times M_2] = [8 \times 8]$, (c) 2-D IAA-I MSC with $[N_{R_1} \times N_{R_2}] = [16 \times 16]$, and (d) 2-D SLIM-MSC.