LUND UNIVERSITY

Computationally Efficient Capon- and APES-based Coherence Spectrum Estimation

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

Published in: IEEE Transactions of Signal Processing

Lund 2012

Mathematical Statistics
Centre for Mathematical Sciences
Lund University

Computationally Efficient Capon- and APES-based Coherence Spectrum Estimation

K. Angelopoulos*, G. O. Glentis*, *Member, IEEE*, and A. Jakobsson†, *Senior Member, IEEE*

***Abstract*— The coherence spectrum is of notable interest as a bivariate spectral measure in a variety of application, and the topic has lately attracted interest with the recent formulation of several high-resolution data adaptive estimators. In this work, we further this development with the presentation of computationally efficient implementations of the Capon- and APES-based MSC estimators. The presented implementations furthers the recent development of exploiting the estimators' inherently low displacement rank of the necessary products of Toeplitz-like matrices to include also the required cross-correlation covariance matrices for the mentioned coherence algorithms. Numerical simulations together with theoretical complexity measures illustrate the performance of the proposed implementations.**

***Index Terms*— Coherence spectrum, data adaptive estimators, efficient algorithms**

I. Introduction

ESTIMATING the coherence between two or more measured signals is a ubiquitous problem, finding applications in a variety of fields, such as speech processing, time series analysis, geophysics, biomedical engineering, and synthetic aperture radar imaging. The topic has lately attracted renewed interest with the proposal of the non-parametric data-dependent Capon-based magnitude squared coherence (MSC) estimator proposed in [1], and then further explored in [2]–[6]. The one- and two-dimensional (2-D) Capon- and APES-based approaches introduced in [1], [2] show that these estimators allow for proper 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 this paper, we further these works by examining both the computational complexity and the performance of the Capon and APES algorithms. Given the high complexity of these methods, we examine ways to form computationally efficient implementations of the algorithms. Reminiscent to the efficient implementations in [7], [8], formulated for the corresponding spectral estimation techniques, these MSC estimators allow for computationally efficient implementations by making use of the inherently low displacement rank of the necessary products of Toeplitz-like matrices, thereby allowing for the development of appropriate Gohberg-Semencul (GS) representations of these matrices. Although the here presented material is related to the results in these earlier works, the GS formulations and relevant data dependent trigonometric polynomials required here for the MSC estimation will differ from those of the corresponding spectral estimators in the formulation of the required cross-spectral density estimates. In the interest of brevity, we will here primarily refer the reader to these earlier works for the details on the implementation of the (auto) spectral densities, and focus on the novel results needed for the MSC estimation. In the following section, we briefly review the Capon- and APES-based MSC estimators, whereafter we proceed to present efficient implementations for the these algorithms. This is done by reformulating the algorithms in terms of data dependent trigonometric polynomials whose kernels are products of Toeplitz-like matrices of various dimensions, which are shown to admit for low displacement rank representations, subsequently utilized for the estimation of the efficient evaluation of the trigonometric polynomials. Then, in Section V, we illustrate the performance of the discussed methods before concluding on the work in Section VI.

This work was supported in part by the Swedish Research Council and Carl Trygger's foundation.

*K. Angelopoulos and G. O. Glentis are with the Department of Science and Technology of Telecommunications, University of Peloponnese, Tripolis, 22100 Greece, email: kaggelop/gglentis@uop.gr.

†A. Jakobsson is with the Dept. of Mathematical Statistics, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden, email: aj@maths.lth.se.

II. 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., [9])

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

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- and APES-based MSC estimates are formed using the matched filter bank framework (see also [9], [10]). Let $\mathbf{h}_M^{(i)} \in C^{M\times 1}$ denote a narrowband data dependent finite impulse response filter centered at a generic frequency $\omega \in [-\pi, \pi]$, and form the signals of interest into $M \times 1$ subvectors

$$\mathbf{x}_M^{(i)}(n) = \begin{bmatrix} x_i(n) & x_i(n-1) & \ldots & x_i(n-M+1) \end{bmatrix}^T \tag{2}$$

for $n = M-1, \ldots, N-1$, where $i = 1$ or 2 for the respective signal, and where $(\cdot)^T$ denotes the transpose. Then, the filter output for the i:th filter at time n is

$$z_i(n) = \mathbf{h}_M^{(i)H}\mathbf{x}_M^{(i)}(n), \tag{3}$$

where $(\cdot)^H$ denotes the conjugate 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.,

$$S_{x_i}(\omega) = E\left\{z_i(n)z_i^H(n)\right\} \approx \mathbf{h}_M^{(i)H}\mathbf{R}_M^{(i)}\mathbf{h}_M^{(i)} \tag{4}$$

where $E\{\cdot\}$ denotes the expectation, and $\mathbf{R}_M^{(i)}$ is an estimate of the signal's covariance matrix, of the form

$$\mathbf{R}_M^{(i)} = \sum_{n=M-1}^{N-1} \mathbf{x}_M^{(i)}(n)\mathbf{x}_M^{(i)H}(n) \tag{5}$$

with $i = 1$ or 2 for the respective signal. Similarly, the cross-spectral density needed to form (1) is estimated as

$$S_{x_1x_2}(\omega) = E\left\{z_1(n)z_2^H(n)\right\} \approx \mathbf{h}_M^{(1)H}\mathbf{R}_M^{(12)}\mathbf{h}_M^{(2)}, \tag{6}$$

with $\mathbf{R}_M^{(12)}$ denoting an estimate of the cross-covariance matrix, given by

$$\mathbf{R}_M^{(12)} = \sum_{n=M-1}^{N-1} \mathbf{x}_M^{(1)}(n)\mathbf{x}_M^{(2)H}(n). \tag{7}$$

As shown in [1], [2], the Capon- and APES-based MSC estimates result from two different design choices for the narrowband filters, yielding the MSC estimates

$$\gamma_{x_1x_2}^{2,Capon}(\omega) = \frac{\left|\mathcal{G}_\omega^{(1)H}\mathbf{R}_M^{12}\mathcal{G}_\omega^{(2)}\right|^2}{\prod_{i=1}^2 \mathbf{f}_M^H(\omega)[\mathbf{R}_M^{(i)}]^{-1}\mathbf{f}_M(\omega)}, \quad (8)$$

$$\gamma_{x_1x_2}^{2,APES}(\omega) = \frac{\left|\breve{\mathcal{G}}_\omega^{(1)H}\mathbf{R}_M^{12}\breve{\mathcal{G}}_\omega^{(2)}\right|^2}{\prod_{i=1}^2 \breve{\mathcal{G}}_\omega^{(i)H}\mathbf{R}_M^{(i)}\breve{\mathcal{G}}_\omega^{(i)}} \quad (9)$$

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

$$\mathbf{f}_M(\omega) = \left[\begin{array}{cccc} 1 & e^{-\jmath\omega} & \dots & e^{-\jmath(M-1)\omega} \end{array}\right]^T \quad (10)$$

whereas the covariance matrices $\mathbf{R}_M^{(1)}$ and $\mathbf{R}_M^{(2)}$, as well as the cross-covariance matrix $\mathbf{R}_M^{(12)}$, are estimated as in (5) and (7), respectively, and the frequency dependent (noise covariance) matrix $\mathbf{Q}_M^{(i)}(\omega)$ is estimated as $\mathbf{Q}_M^{(i)}(\omega) = \mathbf{R}_M^{(i)} - \frac{1}{L}\mathbf{g}_M^{(i)}(\omega)\mathbf{g}_M^{(i)H}(\omega)$, where $L = N - M + 1$, and

$$\mathbf{g}_M^{(i)}(\omega) = \sum_{n=M-1}^{N-1} \mathbf{x}_M^{(i)}(n)e^{-\jmath\omega n}. \quad (11)$$

Direct, brute force, computation of the resulting Capon- and APES-based MSC estimates require a notable amount of computations. Assuming a uniformly spaced frequency grid with K grid points, they require approximately $\mathcal{C}^{Capon} \approx 2M^3 + (3L + 3K)M^2$ and $\mathcal{C}^{APES} \approx \left(2M^3 + 7M^2 + 2LM\right)K$ operations.

III. MSC COMPUTATION USING DATA DEPENDENT TRIGONOMETRIC POLYNOMIALS

To reduce the computational complexity of forming the discussed MSC estimators, we examine how one may exploit the matrix structure to reduce the amount of necessary computations notably. To do so, we begin with examining the Capon-based estimator, noting that (8) may be reformulated in terms of data dependent trigonometric polynomials as

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

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

$$\varphi_i(\omega) \triangleq \mathbf{f}_M^H(\omega)[\mathbf{R}_M^{(i)}]^{-1}\mathbf{f}_M(\omega), \quad (13)$$

for $i = 1$ or 2, and

$$\varphi_{12}(\omega) \triangleq \mathbf{f}_M^H(\omega)\mathbf{P}_M\mathbf{f}_M(\omega) \quad (14)$$

$$\mathbf{P}_M \triangleq [\mathbf{R}_M^{(1)}]^{-1}\mathbf{R}_M^{(12)}[\mathbf{R}_M^{(2)}]^{-1}. \quad (15)$$

To form the APES-based MSC estimate, rewrite (11) as

$$\mathbf{g}_M^{(i)}(\omega) = \mathbf{X}_{M,L}^{(i)}\mathbf{f}_L(\omega)e^{-\jmath\omega(M-1)}. \quad (16)$$

where

$$\mathbf{X}_{M,L}^{(i)} = \left[\begin{array}{cccc} \mathbf{x}_M^{(i)}(M-1) & \mathbf{x}_M^{(i)}(M) & \dots & \mathbf{x}_M^{(i)}(N) \end{array}\right] \quad (17)$$

which, after some algebraic manipulation, allows (9) to be expressed in terms of trigonometric polynomials as (18), given at the top of the next page, where

$$\nu_i(\omega) = \mathbf{f}_M^H(\omega)\mathbf{G}_{M,L}^{(i)}\mathbf{f}_L(\omega) \quad (19)$$

$$\xi_i(\omega) = \mathbf{f}_L^H(\omega)\mathbf{H}_{L,L}^{(i)}\mathbf{f}_L(\omega) \quad (20)$$

$$\nu_{12}(\omega) = \mathbf{f}_M^H(\omega)\mathbf{G}_{M,L}^{(12)}\mathbf{f}_L(\omega) \quad (21)$$

$$\nu_{21}(\omega) = \mathbf{f}_M^H(\omega)\mathbf{G}_{M,L}^{(21)}\mathbf{f}_L(\omega) \quad (22)$$

$$\xi_{12}(\omega) = \mathbf{f}_L^H(\omega)\mathbf{H}_{L,L}^{(12)}\mathbf{f}_L(\omega) \quad (23)$$

with data adaptive kernels defined as

$$\mathbf{G}_{M,L}^{(i)} = [\mathbf{R}_M^{(i)}]^{-1}\mathbf{X}_{M,L}^{(i)} \quad (24)$$

$$\mathbf{H}_{L,L}^{(i)} = \mathbf{X}_{M,L}^{(i)H}[\mathbf{R}_M^{(i)}]^{-1}\mathbf{X}_{M,L}^{(i)} \quad (25)$$

$$\mathbf{G}_{M,L}^{(12)} = \mathbf{P}_M\mathbf{X}_{M,L}^{(1)} \quad (26)$$

$$\mathbf{G}_{M,L}^{(21)} = \mathbf{P}_M^H\mathbf{X}_{M,L}^{(2)} \quad (27)$$

$$\mathbf{H}_{L,L}^{(12)} = \mathbf{X}_{M,L}^{(1)H}\mathbf{P}_M\mathbf{X}_{M,L}^{(2)} \quad (28)$$

The advantage of using trigonometric polynomials in the MSC estimation stems from the fact that these can efficiently be evaluated on the unit circle using the Fast Fourier Transform (FFT), since for a kernel $\mathbf{A}_{M,L}$ of dimensions $M \times L$, one may have

$$\varphi(\omega) \triangleq \mathbf{f}_M^H(\omega)\mathbf{A}_{M,L}\mathbf{f}_L(\omega) = \sum_{\kappa=-M+1}^{L-1} c_\kappa e^{\jmath k\omega}. \quad (29)$$

When $\mathbf{A}_{M,L}$ is available, the coefficients of the trigonometric polynomial c_κ can be estimated by adding up the matrix elements upon the diagonals. Direct estimation of the kernels associated with the data adaptive trigonometric polynomials required for the computation of the Capon-based MSC, i.e., (13) and (14), and the APES-based MSC, i.e., (19)-(23), results in an unnecessarily high computational complexity, since the pertinent kernels are products of Toeplitz-like matrices thus being Toeplitz-like matrices themselves and as shown in [7], [11], trigonometric polynomials related to Toeplitz-like matrices can directly (and efficiently) be evaluated from their displacement representation bypassing the need of constructing these matrices explicitly. For the readers benefit and for further use, we proceed by presenting briefly the basics from the displacement representation theory of matrices (see also [12]–[15]), recalling that the displacement of the matrix $\mathbf{A}_{M,L} \in \mathcal{C}^{M\times L}$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_L^T$ is defined as

$$\nabla_{Z_M,Z_L^T}\mathbf{A}_{M,L} \triangleq \mathbf{A}_{M,L} - \mathbf{Z}_M\mathbf{A}_{M,L}\mathbf{Z}_{M,L}^T. \quad (30)$$

where $\mathbf{Z}_M$ and $\mathbf{Z}_L$ are lower shifting matrices of dimensions $M\times M$ and $L\times L$, respectively. Suppose that there exist integers ρ and $\sigma_i \in \{-1, 1\}$, $i = 1, 2\dots\rho$, such that

$$\nabla_{Z_M,Z_L^T}\mathbf{A}_{M,L} = \sum_{i=1}^{\rho}\sigma_i\mathbf{t}_M^i\mathbf{s}_L^{iH} = \mathbf{T}_{M,\rho}\mathbf{\Sigma}_\rho\mathbf{S}_{L,\rho}^H \quad (31)$$

where $\mathbf{T}_{M,\rho} = \left[\begin{array}{ccc} \mathbf{t}_M^1 & \dots & \mathbf{t}_M^\rho \end{array}\right]$,$\mathbf{S}_{L,\rho} = \left[\begin{array}{ccc} \mathbf{s}_L^1 & \dots & \mathbf{s}_L^\rho \end{array}\right]$, and $\mathbf{\Sigma}_\rho = diag\left(\sigma_1\dots\sigma_\rho\right)$, whereas $diag(\mathbf{a})$ denotes the diagonal matrix formed with the vector $\mathbf{a}$ along its diagonal, and with $\mathbf{t}_M^i$ and $\mathbf{s}_L^i$ being the so-called generator vectors. The displacement rank of the representation equals the rank of the associated displacement matrix, $\nabla_{Z_M,Z_L^T}\mathbf{A}_{M,L}$, whereas the integer ρ may be larger than or equal to the corresponding displacement rank. In summary, the triplet $(\mathbf{T}_{M,\rho}, \mathbf{S}_{L,\rho}, \mathbf{\Sigma}_\rho)$ is called the displacement representation of $\mathbf{A}_{M,L}$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_L^T$. Then, the GS factorization of $\mathbf{A}_{M,L}$ may be expressed as

$$\mathbf{A}_{M,L} = \sum_{i=1}^{\rho}\sigma_i\mathcal{L}_M(\mathbf{t}_M^i)\mathcal{L}_L^H(\mathbf{s}_L^i) \quad (32)$$

where $\mathcal{L}_M(\mathbf{x})$ ($\mathcal{L}_L(\mathbf{y})$) denotes a Krylov matrix of the form

$$\mathcal{L}(\mathbf{x}) = \left[\begin{array}{cccc} \mathbf{x} & Z_M\mathbf{x} & Z_M^2\mathbf{x} & \dots & Z_M^{M-1}\mathbf{x} \end{array}\right]. \quad (33)$$

Given the displacement representation of $\mathbf{A}_{M,L}$ the coefficients of the associated trigonometric polynomial (29) can be efficiently computed directly from the generator vectors bypassing the need of forming $\mathbf{A}_{M,L}$ explicitly, [7], [11]. Moreover, using the displacement representation of $\mathbf{A}_{M,L}$, related matrix vector products can be computed in a fast manner using the FFT.

$$\gamma_{x_1x_2}^{2,APES}(\omega)=\frac{|(L-\xi_1(\omega))(L-\xi_2(\omega))\varphi_{12}(\omega)+(L-\xi_1(\omega))\nu_{21}(\omega)\nu_2^*(\omega)+(L-\xi_2(\omega))\nu_{12}^*(\omega)\nu_1(\omega)+\nu_1(\omega)\xi_{12}(\omega)\nu_2^*(\omega)|^2}{\prod_{i=1}^2\left((L-\xi_i(\omega))^2\varphi_i(\omega)+(2L-\xi_i(\omega))|\nu_i(\omega)|^2\right)} \quad (18)$$

IV. Displacement representation of data dependent kernels

The efficient evaluation of the data dependent trigonometric polynomials in (13) and (14), as well as (19)-(23), relies upon the computation of the displacement representation of the pertinent kernels. Thanks to the Toeplitz-like structure of the data covariance matrices (5), a displacement representation of their inverses $[\mathbf{R}_M^{(i)}]^{-1}$, for $i=1$ and $i=2$, can be efficiently computed and subsequently be utilized for the computation of the displacement representation of $\mathbf{G}_{M,L}^{(i)}$ and $\mathbf{H}_{L,L}^{(i)}$, defined by (24) and (25), respectively, as it has been recently shown in [7]. We further these results by developing novel and efficient displacement representations for the remaining kernels involved in the fast computation of the Capon and APES MSC using data dependent trigonometric polynomials, namely (15) and (26)-(28). We proceed to present the key properties of the covariance matrices as well as to define the basic parameters associated with the displacement generators of the pertinent kernels. Consider the order partitions of the data vectors

$$\mathbf{x}_M^{(i)}(n)=\left[\begin{array}{c}\mathbf{x}_{M-1}^{(i)}(n)\\ x^{(i)}(n-M+1)\end{array}\right]=\left[\begin{array}{c}x^{(i)}(n)\\ \mathbf{x}_{M-1}^{(i)}(n-1)\end{array}\right] \quad (34)$$

implying that $\mathbf{R}_M^{(i)}$ may be partitioned as

$$\mathbf{R}_M^{(i)}=\left[\begin{array}{cc}\mathbf{R}_{M-1}^{(i)} & \mathbf{r}_{M-1}^{b(i)}\\ \mathbf{r}_{M-1}^{b(i)H} & r_M^{bo(i)}\end{array}\right]=\left[\begin{array}{cc}r_M^{fo(i)} & \mathbf{r}_{M-1}^{f(i)H}\\ \mathbf{r}_{M-1}^{f(i)} & \widehat{\mathbf{R}}_{M-1}^{(i)}\end{array}\right], \quad (35)$$

with the resulting forward and backward predictors and the associated prediction powers being defined as

$$\bar{\mathbf{a}}_M^{(i)}=\left[\begin{array}{c}1\\ \mathbf{a}_{M-1}^{(i)}\end{array}\right], \quad \bar{\mathbf{b}}_M^{(i)}=\left[\begin{array}{c}\mathbf{b}_{M-1}^{(i)}\\ 1\end{array}\right] \quad (36)$$

and

$$\begin{aligned}\mathbf{a}_{M-1}^{(i)} &= -[\widehat{\mathbf{R}}_{M-1}^{(i)}]^{-1}\mathbf{r}_{M-1}^{f(i)}, &(37)\\ \mathbf{b}_{M-1}^{(i)} &= -[\mathbf{R}_{M-1}^{(i)}]^{-1}\mathbf{r}_{M-1}^{b(i)}, &(38)\\ \alpha_M^{f(i)} &= r_M^{fo(i)}+\mathbf{r}_{M-1}^{f(i)H}\mathbf{a}_{M-1}^{(i)}, &(39)\\ \alpha_M^{b(i)} &= r_M^{bo(i)}+\mathbf{r}_{M-1}^{b(i)H}\mathbf{b}_{M-1}^{(i)}. &(40)\end{aligned}$$

Furthermore, define the rank-two modification of the (lower order) covariance matrix as

$$\begin{aligned}\tilde{\mathbf{R}}_{M-1}^{(i)} &= \mathbf{R}_{M-1}^{(i)}-\mathbf{x}_{M-1}^{(i)}(N-1)\mathbf{x}_{M-1}^{(i)H}(N-1), &(41)\\ \widehat{\mathbf{R}}_{M-1}^{(i)} &= \tilde{\mathbf{R}}_{M-1}^{(i)}+\mathbf{x}_{M-1}^{(i)}(M-2)\mathbf{x}_{M-1}^{(i)H}(M-2). &(42)\end{aligned}$$

and the resulting Kalman type vectors and their powers as

$$\bar{\mathbf{w}}_M^{(i)}=\left[\begin{array}{c}0\\ \mathbf{w}_{M-1}^{(i)}\end{array}\right], \quad \bar{\mathbf{v}}_M^{(i)}=\left[\begin{array}{c}0\\ \mathbf{v}_{M-1}^{(i)}\end{array}\right] \quad (43)$$

and

$$\begin{aligned}\mathbf{w}_{M-1}^{(i)} &= -[\tilde{\mathbf{R}}_{M-1}^{(i)}]^{-1}\mathbf{x}_{M-1}^{(i)}(N-1), &(44)\\ \mathbf{v}_{m-1}^{(i)} &= -[\widehat{\mathbf{R}}_{m-1}^{(i)}]^{-1}\mathbf{x}_{m-1}^{(i)}(M-2), &(45)\\ \alpha_M^{w(i)} &= 1-\mathbf{x}_{M-1}^{(i)H}(N-1)\mathbf{w}_{M-1}^{(i)}, &(46)\\ \alpha_M^{v(i)} &= 1+\mathbf{x}_{M-1}^{(i)H}(M-2)\mathbf{v}_{M-1}^{(i)}. &(47)\end{aligned}$$

Combining these results, a displacement representation of $[\mathbf{R}_M^{(1)}]^{-1}$ and $[\mathbf{R}_M^{(2)}]^{-1}$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_M^T$ with displacement rank of $\rho^R=4$ has been presented in [7] and subsequently been utilized for the efficient evaluation of the trigonometric polynomials $\varphi_i(\omega)$ in (13). Given the GS representations of $\mathbf{R}_M^{-1(1)}$ and $\mathbf{R}_M^{-1(2)}$, we proceed to form a representation of the product that appears in (14), i.e.,

$$\mathbf{P}_M \triangleq \mathbf{R}_M^{-1(1)}\mathbf{R}_M^{(12)}\mathbf{R}_M^{-1(2)}. \quad (48)$$

As the displacement rank of $\mathbf{R}_M^{-1(1)}$ and $\mathbf{R}_M^{-1(2)}$ is $\rho^{R1}=\rho^{R2}=4$, and the displacement rank of $\mathbf{R}_M^{(12)}$ is $\rho^{R12}=4$, the displacement rank of the product of these three matrices may be expected to be $\rho^P\leq 14$ (see also [12], [14]), although, as is proved in Appendix A, one may in fact form a more compact displacement representation as:

Lemma 1: A displacement representation of $\mathbf{P}_M$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_M^T$ may be formed as $\left(\mathbf{\Upsilon}_{M,8},\mathbf{\Psi}_{M,8},\mathbf{\Sigma}_8^P\right)$, where

$$\begin{aligned}\mathbf{\Upsilon}_{M,8} &= \left[\begin{array}{ccc}\boldsymbol{\upsilon}_M^1 & \dots & \boldsymbol{\upsilon}_M^8\end{array}\right], &(49)\\ \mathbf{\Psi}_{M,8} &= \left[\begin{array}{ccc}\boldsymbol{\psi}_M^1 & \dots & \boldsymbol{\psi}_M^8\end{array}\right], &(50)\\ \mathbf{\Sigma}_8^P &= diag(\sigma_1^P,\dots\sigma_8^P) &(51)\end{aligned}$$

TABLE I
Auxiliary variables required for the displacement representation of $\mathbf{P}_M$.

$$\begin{aligned}c_M^o &= \bar{\mathbf{b}}_M^{(1)H}\mathbf{R}_M^{(12)}\bar{\mathbf{b}}_M^{(2)}\\ \mathbf{c}_M^{(1)} &= \left[\begin{array}{cc}[\mathbf{R}_{M-1}^{(2)}]^{-1} & \mathbf{0}\\ \mathbf{0}^T & 0\end{array}\right]\mathbf{R}_M^{(12)H}\bar{\mathbf{b}}_M^{(1)}\\ \mathbf{c}_M^{(2)} &= \left[\begin{array}{cc}[\mathbf{R}_{M-1}^{(1)}]^{-1} & \mathbf{0}\\ \mathbf{0}^T & 0\end{array}\right]\mathbf{R}_M^{(12)}\bar{\mathbf{b}}_M^{(2)}\\ \mathbf{c}_M^{(12)} &= \mathbf{c}_M^{(1)}+\bar{\mathbf{b}}_M^{(2)}\frac{c_M^{o*}}{\alpha_M^{b(2)}}\\ d_M^o &= \bar{\mathbf{a}}_M^{(1)H}\mathbf{R}_M^{(12)}\bar{\mathbf{a}}_M^{(2)}\\ \mathbf{d}_M^{(1)} &= \left[\begin{array}{cc}0 & \mathbf{0}^T\\ \mathbf{0} & [\widehat{\mathbf{R}}_{M-1}^{(2)}]^{-1}\end{array}\right]\mathbf{R}_M^{(12)H}\bar{\mathbf{a}}_M^{(1)}\\ \mathbf{d}_M^{(2)} &= \left[\begin{array}{cc}0 & \mathbf{0}^T\\ \mathbf{0} & [\widehat{\mathbf{R}}_{M-1}^{(1)}]^{-1}\end{array}\right]\mathbf{R}_M^{(12)}\bar{\mathbf{a}}_M^{(2)}\\ \mathbf{d}_M^{(12)} &= \mathbf{d}_M^{(1)}+\bar{\mathbf{a}}_M^{(2)}\frac{d_M^{o*}}{\alpha_M^{f(2)}}\\ p_M^o &= \mathbf{w}_{M-1}^{(1)H}\tilde{\mathbf{R}}_{M-1}^{(12)}\mathbf{w}_{M-1}^{(2)}\\ \mathbf{p}_{M-1}^{(1)} &= [\tilde{\mathbf{R}}_{M-1}^{(1)}]^{-1}\tilde{\mathbf{R}}_{M-1}^{(12)}\mathbf{w}_{M-1}^{(2)}-\frac{\mathbf{w}_{M-1}^{(1)}}{\alpha_M^{w(1)}}\\ \mathbf{p}_{M-1}^{(2)} &= [\tilde{\mathbf{R}}_{M-1}^{(2)}]^{-1}\tilde{\mathbf{R}}_{M-1}^{(12)H}\mathbf{w}_{M-1}^{(1)}-\mathbf{w}_{M-1}^{(2)}\frac{p_M^{o*}}{\alpha_M^{w(2)}}\\ q_M^o &= \mathbf{v}_{M-1}^{(1)H}\hat{\mathbf{R}}_{M-1}^{(12)}\mathbf{v}_{M-1}^{(2)}\\ \mathbf{q}_{M-1}^{(1)} &= [\hat{\mathbf{R}}_{M-1}^{(1)}]^{-1}\hat{\mathbf{R}}_{M-1}^{(12)}\mathbf{v}_{M-1}^{(2)}-\frac{\mathbf{v}_{M-1}^{(1)}}{\alpha_M^{v(1)}}\\ \mathbf{q}_{M-1}^{(2)} &= [\hat{\mathbf{R}}_{M-1}^{(2)}]^{-1}\hat{\mathbf{R}}_{M-1}^{(12)H}\mathbf{v}_{M-1}^{(1)}+\mathbf{v}_{M-1}^{(2)}\frac{q_M^{o*}}{\alpha_M^{v(2)}}\\ \bar{\mathbf{p}}_M^{(i)} &= \left[\begin{array}{c}0\\ \mathbf{p}_{M-1}^{(i)}\end{array}\right], \quad \bar{\mathbf{q}}_M^{(i)}=\left[\begin{array}{c}0\\ \mathbf{q}_{M-1}^{(i)}\end{array}\right], \quad i=1,2\end{aligned}$$

with the set of auxiliary variables tabulated in Table I, and

$$\boldsymbol{v}_M^1 = \bar{\mathbf{a}}_M^{(1)}/\sqrt{\alpha}_M^{f(1)}, \quad \boldsymbol{\psi}_M^1 = \mathbf{d}_M^{(12)}/\sqrt{\alpha}_M^{f(1)}, \quad \sigma_1^P = 1 \tag{52}$$

$$\boldsymbol{v}_M^2 = \mathbf{d}_M^{(2)}/\sqrt{\alpha}_M^{f(2)}, \quad \boldsymbol{\psi}_M^2 = \bar{\mathbf{a}}_M^{(2)}/\sqrt{\alpha}_M^{f(2)}, \quad \sigma_2^P = 1 \tag{53}$$

$$\boldsymbol{v}_M^3 = \mathbf{Z}_M\bar{\mathbf{b}}_M^{(1)}/\sqrt{\alpha}_M^{b(1)}, \quad \boldsymbol{\psi}_M^3 = \mathbf{Z}_M\mathbf{c}_M^{(12)}/\sqrt{\alpha}_M^{b(1)}, \quad \sigma_3^P = -1 \tag{54}$$

$$\boldsymbol{v}_M^4 = \mathbf{Z}_M\mathbf{c}_M^{(2)}/\sqrt{\alpha}_M^{b(2)}, \quad \boldsymbol{\psi}_M^4 = \mathbf{Z}_M\bar{\mathbf{b}}_M^{(2)}/\sqrt{\alpha}_M^{b(2)}, \quad \sigma_4^P = -1 \tag{55}$$

$$\boldsymbol{v}_M^5 = \bar{\mathbf{w}}_M^{(1)}/\sqrt{\alpha}_M^{w(1)}, \quad \boldsymbol{\psi}_M^5 = \bar{\mathbf{p}}_M^{(2)}/\sqrt{\alpha}_M^{w(1)}, \quad \sigma_5^P = 1 \tag{56}$$

$$\boldsymbol{v}_M^6 = \bar{\mathbf{p}}_M^{(1)}/\sqrt{\alpha}_M^{w(2)}, \quad \boldsymbol{\psi}_M^6 = \bar{\mathbf{w}}_M^{(2)}/\sqrt{\alpha}_M^{w(2)}, \quad \sigma_6^P = 1 \tag{57}$$

$$\boldsymbol{v}_M^7 = \bar{\mathbf{v}}_M^{(1)}/\sqrt{\alpha}_M^{v(1)}, \quad \boldsymbol{\psi}_M^7 = \bar{\mathbf{q}}_M^{(2)}/\sqrt{\alpha}_M^{v(1)}, \quad \sigma_7^P = -1 \tag{58}$$

$$\boldsymbol{v}_M^8 = \bar{\mathbf{q}}_M^{(1)}/\sqrt{\alpha}_M^{v(2)}, \quad \boldsymbol{\psi}_M^8 = \bar{\mathbf{v}}_M^{(2)}/\sqrt{\alpha}_M^{v(2)}, \quad \sigma_8^P = -1. \tag{59}$$

The displacement rank of the representation is $\rho^P = 8$. □

To complete the derivation of the fast algorithm for the estimation of the displacement representation of $\mathbf{P}_M$, several matrix vector products associated with the variables that appear in Table I should be organized in an efficient way, which is in fact feasible, since one can show that:

Lemma 2: Given the displacement representation of $[\mathbf{R}_M^{(i)}]^{-1}$, as in Lemma 3 of [7], the displacement representation of

$$\begin{bmatrix} [\mathbf{R}_{M-1}^{(i)}]^{-1} & 0 \\ 0^T & 0 \end{bmatrix}$$

with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_M^T$ may be formed as $\left(\tilde{\mathbf{T}}_{M,4}^{(i)}, \tilde{\mathbf{T}}_{M,4}^{(i)}, \Sigma_4^R\right)$, where

$$\tilde{\mathbf{T}}_{M,4}^{(i)} = \begin{bmatrix} \mathbf{t}_M^{(i),1} & \tilde{\mathbf{t}}_M^{(i),2} & \mathbf{t}_M^{(i),3} & \mathbf{t}_M^{(i),4} \end{bmatrix}, \tag{60}$$

$$\Sigma_4^R = diag(\sigma_1^R, \ldots \sigma_4^R) \tag{61}$$

for $i = 1, 2$ and with $\mathbf{t}_M^{(i),1} = \bar{\mathbf{a}}_M^{(i)}/\sqrt{\alpha}_M^{f(i)}$, $\sigma_1^R = 1$, $\tilde{\mathbf{t}}_M^{(i),2} = \bar{\mathbf{b}}_M^{(i)}/\sqrt{\alpha}_M^{b(i)}$, $\sigma_2^R = -1$, $\mathbf{t}_M^{(i),3} = \bar{\mathbf{w}}_M^{(i)}/\sqrt{\alpha}_M^{w(i)}$, $\sigma_3^R = 1$, $\sigma_4^R = -1$, and $\mathbf{t}_M^{(i),4} = \bar{\mathbf{v}}_M/\sqrt{\alpha}_M^{v(i)}$. □

Given the displacement representation of $\mathbf{R}_M^{-1(i)}$, the displacement representation of $\mathbf{G}_{M,L}^{(i)}$ and $\mathbf{H}_{L,L}^{(i)}$, as defined by (24) and (25), can be computed using a set of auxiliary variables defined as

$$\mathbf{e}_L^{a(i)} = \mathbf{X}_{M,L}^{(i)H}\bar{\mathbf{a}}_M^{(i)}, \quad \mathbf{e}_L^{b(i)} = \mathbf{X}_{M,L}^{(i)H}\bar{\mathbf{b}}_M^{(i)} \tag{62}$$

$$\mathbf{e}_L^{w(i)} = \mathbf{X}_{M,L}^{(i)H}\bar{\mathbf{w}}_M^{(i)}, \quad \mathbf{e}_L^{v(i)} = \mathbf{X}_{M,L}^{(i)H}\bar{\mathbf{v}}_M^{(i)}, \tag{63}$$

whereby the polynomials $\varphi_i(\omega)$, $\nu_i(\omega)$, and $\xi_i(\omega)$, for $i = 1, 2$, as defined by (13), (19) and (20), respectively, can be computed efficiently using the method presented in [7], while $\varphi_{12}(\omega)$ may be computed as discussed above. Thus, it remains to show how the polynomials $\nu_{12}(\omega)$, $\nu_{21}(\omega)$, and $\xi_{12}(\omega)$, defined by (14) and (21)-(23), respectively, can be computed efficiently. To do so, we make use of Lemma 1 to form displacement representations of the matrices $\mathbf{G}_{M,L}^{(12)}$ and $\mathbf{G}_{M,L}^{(21)}$, given by (26) and (27), and we introduce a further set of auxiliary variables, similar to those defined by (62) and (63), as

$$\mathbf{e}_L^{p(2)} = \mathbf{X}_{M,L}^{(2)H}\bar{\mathbf{p}}_M^{(2)}, \quad \mathbf{e}_L^{p(1)} = \mathbf{X}_{M,L}^{(1)H}\bar{\mathbf{p}}_M^{(1)} \tag{64}$$

$$\mathbf{e}_L^{q(2)} = \mathbf{X}_{M,L}^{(2)H}\bar{\mathbf{q}}_M^{(2)}, \quad \mathbf{e}_L^{q(1)} = \mathbf{X}_{M,L}^{(1)H}\bar{\mathbf{q}}_M^{(1)} \tag{65}$$

$$\mathbf{e}_L^{c(12)} = \mathbf{X}_L^{(2)H}\mathbf{c}_M^{(12)}, \quad \mathbf{e}_L^{c(1)} = \mathbf{X}_{M,L}^{(1)H}\mathbf{c}_M^{(2)} \tag{66}$$

$$\mathbf{e}_L^{d(12)} = \mathbf{X}_{M,L}^{(2)H}\mathbf{d}_M^{(12)}, \quad \mathbf{e}_L^{d(1)} = \mathbf{X}_{M,L}^{(1)H}\mathbf{d}_M^{(2)} \tag{67}$$

Reminiscing the approach following in the derivation of the displacement representation presented in Lemma 1, and after some further algebraic manipulation, a displacement representation of $\mathbf{G}_{M,L}^{(21)}$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_L^T$ may then be formed as $\left(\boldsymbol{\Upsilon}_{M,8}, \Xi_{L,8}^{(2)}, \Sigma_8^P\right)$, where

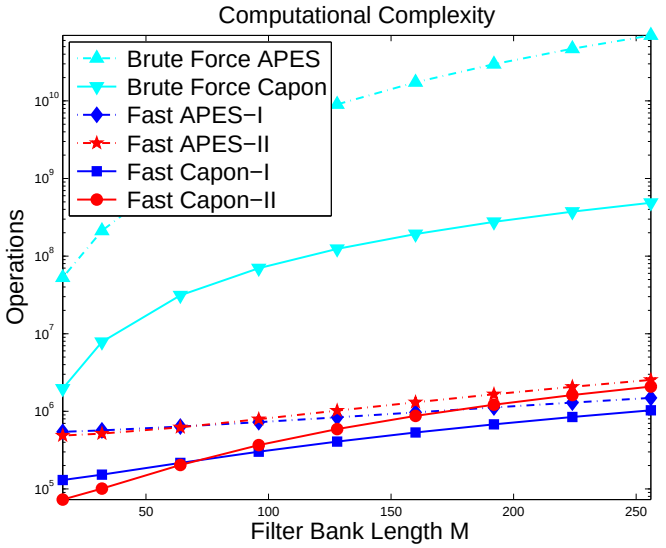


Fig. 1. Computational complexity of the Capon- and APES-based algorithms using the proposed and the brute force implementations, for varying filter lengths, M, for $N = 512$, and with $K = 2048$.

$$\Xi_{L,8}^{(2)} = \begin{bmatrix} \boldsymbol{\xi}_L^{2,1} & \ldots & \boldsymbol{\xi}_L^{2,8} \end{bmatrix}, \tag{68}$$

with

$$\begin{aligned}
\boldsymbol{\xi}_L^{2,1} &= \mathbf{e}_L^{d(12)}/\sqrt{\alpha}_M^{f(1)}, \quad \boldsymbol{\xi}_L^{2,2} = \mathbf{e}_L^{a(2)}/\sqrt{\alpha}_M^{f(2)} \\
\boldsymbol{\xi}_L^{2,3} &= \mathbf{Z}_L\mathbf{e}_L^{c(12)}/\sqrt{\alpha}_M^{b(1)}, \quad \boldsymbol{\xi}_L^{2,4} = \mathbf{Z}_L\mathbf{e}_L^{b(2)}/\sqrt{\alpha}_M^{b(2)} \\
\boldsymbol{\xi}_L^{2,5} &= \mathcal{P}[\mathbf{e}_L^{p(2)}]^{[0]}/\sqrt{\alpha}_M^{w(1)}, \quad \boldsymbol{\xi}_L^{2,6} = \mathcal{P}[\mathbf{e}_L^{w2}]^{[0]}/\sqrt{\alpha}_M^{w(2)} \\
\boldsymbol{\xi}_L^{2,7} &= \mathcal{P}[\mathbf{e}_L^{q(2)}]^{[1]}/\sqrt{\alpha}_M^{v(1)}, \quad \boldsymbol{\xi}_L^{2,8} = \mathcal{P}[\mathbf{e}_L^{v(2)}]^{[\alpha_M^{v(2)}]}/\sqrt{\alpha}_M^{v(2)}
\end{aligned}$$

Here, the operator $\mathcal{P}[\mathbf{x}]^{[\chi]}$ replaces the first element of the vector upon which it acts by χ, i.e., if $\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \ldots & x_{L-1} \end{bmatrix}^T$, then

$$\mathcal{P}[\mathbf{x}]^{[\chi]} = \begin{bmatrix} \chi & x_1 & \ldots & x_{L-1} \end{bmatrix}^T. \tag{69}$$

Similarly, a displacement representation of $\mathbf{G}_{M,L}^{(12)}$ with respect to $\mathbf{Z}_M$ and $\mathbf{Z}_L^T$ may be formed as $\left(\boldsymbol{\Psi}_{M,8}, \Xi_{L,8}^{(1)}, \Sigma_8^P\right)$, where

$$\Xi_{L,8}^{(1)} = \begin{bmatrix} \boldsymbol{\xi}_L^{1,1} & \ldots & \boldsymbol{\xi}_L^{1,8} \end{bmatrix} \tag{70}$$

with

$$\begin{aligned}
\boldsymbol{\xi}_L^{1,1} &= \mathbf{e}_L^{a(1)}/\sqrt{\alpha}_M^{f(1)}, \quad \boldsymbol{\xi}_{\mathcal{L}}^{1,2} = \mathbf{e}_L^{d(1)}/\sqrt{\alpha}_M^{f(2)}, \\
\boldsymbol{\xi}_L^{1,3} &= \mathbf{Z}_L\mathbf{e}_L^{b(1)}/\sqrt{\alpha}_M^{b(1)}, \quad \boldsymbol{\xi}_L^{1,4} = \mathbf{Z}_L\mathbf{e}_L^{c(1)}/\sqrt{\alpha}_M^{b(2)}, \\
\boldsymbol{\xi}_L^{1,5} &= \mathcal{P}[\mathbf{e}_L^{w(1)}]^{[0]}/\sqrt{\alpha}_M^{w(1)}, \quad \boldsymbol{\xi}_L^{1,6} = \mathcal{P}[\mathbf{e}_L^{p(1)}]^{[0]}/\sqrt{\alpha}_M^{w(2)}, \\
\boldsymbol{\xi}_L^{1,7} &= \mathcal{P}[\mathbf{e}_L^{v(1)}]^{[\alpha_M^{v1}]}/\sqrt{\alpha}_M^{v(1)}, \quad \boldsymbol{\xi}_L^{1,8} = \mathcal{P}[\mathbf{e}_L^{q(1)}]^{[-q_M^o]}/\sqrt{\alpha}_M^{v(2)},
\end{aligned}$$

Finally, a displacement representation of $\mathbf{H}_{L,L}^{12}$ with respect to $\mathbf{Z}_L$ and $\mathbf{Z}_L^T$ may be formed as $\left([\Xi_{L,8}^{(1)}\ \mathbf{1}_L], [\Xi_{L,8}^{(2)}\ \mathbf{1}_L], (\Sigma_8^P, 1)\right)$, where $\mathbf{1}_L$ is a $L \times 1$ vector with a one in the first position followed by $L-1$ zeros. The displacement rank of the matrix $\mathbf{H}_{L,L}^{12}$ is $\rho^{H12} = 9$. The derivation of the proposed displacement representation of $\mathbf{H}_{L,L}^{(12)}$ is supplied in Appendix B, noting that the displacement representations of $\mathbf{G}_{M,L}^{(12)}$ and $\mathbf{G}_{M,L}^{(21)}$ are derived in a similar way.

A. Overall organization

An efficient implementation of the Capon-based MSC estimate may thus be formed by first computing the displacement representation of $[\mathbf{R}_M^{(i)}]^{-1}$, for $i = 1, 2$, using the generalized Levinson algorithm for the computation of the pertinent variables, as given in (37)-(40) and (44)-(47) [7]. The computational complexity of this step is $9M^2 + 6\phi(N)$, where $\phi(N)$ denotes the cost for the computation of the FFT. Then, using the displacement representation of $[\mathbf{R}_M^{(i)}]^{-1}$, for $i = 1, 2$, compute the displacement representation of $\mathbf{P}_M$ using Lemma 1, where the pertinent variables in (52)-(59) are estimated using Table I and Lemma 2 at a cost of $20M^2 + 6\phi(N)$ operations, if conventional matrix vector multiplication is applied, or at a cost of $88\phi(2M) + 26\phi(N)$ when fast Toeplitz vector multiplication using (32) is used. Using these representations, one may then compute the coefficients of the trigonometric polynomials required to form (13) and (14), respectively, using the results in [7], at a cost of $52\phi(M)$. These polynomials are then evaluated over a uniform grid of size K using the FFT, yielding the Capon-based MSC estimate via (12) at a cost of $3\phi(K)$ operations. Thus, the total computational complexity of the fast Capon-based MSC implementation is

$$\begin{aligned} C^{FCapon-I} &\approx 9M^2 + 140\phi(2M) + 32\phi(N) + 3\phi(K) \\ C^{FCapon-II} &\approx 29M^2 + 52\phi(2M) + 12\phi(N) + 3\phi(K) \end{aligned}$$

where the first method is the one using the fast Toeplitz vector multiplication, whereas the latter use the conventional matrix vector multiplication. Thus, the first method is asymptotically faster that the second one, whereas, for short filter bank lengths M, the second method is faster. The APES-based MSC is computationally more intensive as compared to the Capon-based counterpart, since on top of the operations required by the Capon-based method, additional computations are necessary for the formulation of the auxiliary variables (62)-(67) and the trigonometric polynomials (19)-(23), all of which may, as shown in [7], be formed using fast convolution and fast trigonometric polynomial evaluation methods based on the FFT. The computational complexity of the proposed fast APES-based MSC estimation method is given by

$$\mathcal{C}^{FAPES} \approx \mathcal{C}^{FCapon} + 52\phi(2M) + 27\phi(2L) + 86\phi(N) + 7\phi(K)$$

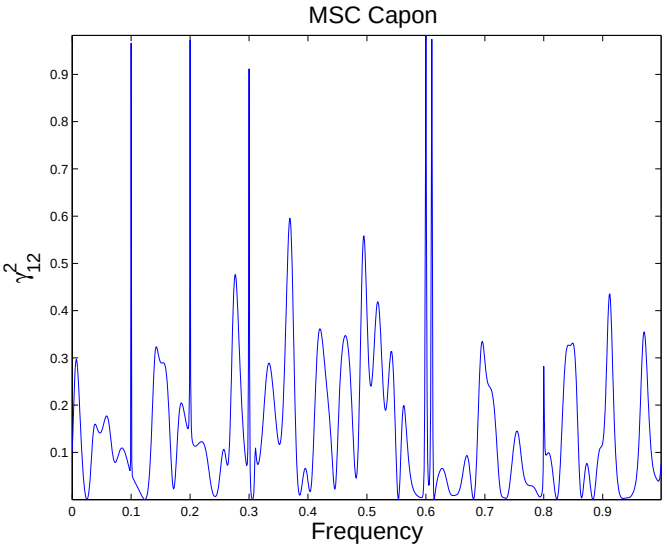
with $\mathcal{C}^{FCapon}$ denoting the complexity of either of the Capon-formulations. The computational complexities of the proposed Capon-based and APES-based methods are illustrated in Fig. 1, for $N = 512$ and for M varying up to $N/2-1$, with $K = 2048$. Clearly, the proposed implementations are up to five orders of magnitude faster than their brute force counterparts. Moreover, despite the fact that the fast implementation of the APES-based MSC is by far more involved than the Capon-based formulation, the complexity overhead is almost marginal asymptotically.

V. Numerical examples

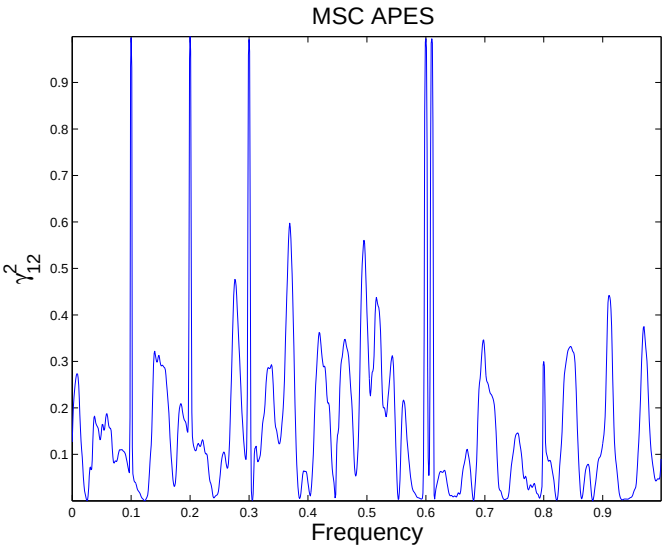
The performance of the proposed MSC estimation algorithms are illustrated by means of computer simulations. Consider $N = 200$ samples of two signals, $x_1(n)$ and $x_2(n)$, which are both a mixture of sinusoidal signals corrupted by additive noise

$$x_i(n) = \sum_{\ell=1}^{7} r_\ell^i e^{\jmath 2\pi f_\ell^i n} + w_i(n), \quad i = 1, 2 \tag{71}$$

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, whereas the Signal to Noise ratio (SNR) is fixed to SNR=5dB. Here, the signals' frequencies are selected as $f^1 = [0.1, 0.2, 0.3, 0.31, 0.6, 0.61, 0.8]$ and $f^2 = [0.1, 0.2, 0.3, 0, 0.6, 0.61, 0]$. The MSC is evaluated over $K = 1000$ uniformly distributed frequency grid points. The Capon- and APES-based MSC estimates are illustrated in Fig. 2(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.



(a)



(b)

Fig. 2. MSC estimation of two cisoidal mixtures using $N = 200$, at SNR=5dB, with $K = 1000$ uniformly spaced frequency points: (a) Capon- and (b) APES-based MSC with filter lengths set equal to $M = 40$,

VI. Conclusions

This works examines the performance of, as well as introduce computationally efficient implementations for, the Capon, and APES-based MSC estimators.The estimators are data-adaptive filter bank formulations of the MSC spectrum and as it has been show they can expressed in terms of data adaptive trigonometric polynomials, whose kernels are products of Toeplitz-like matrices, allowing for low rank displacement representation, which is subsequently utilized for the efficient computation of the polynomial coefficients and

$$\mathbf{P}_M - \mathbf{Z}_M\mathbf{P}_M\mathbf{Z}_M^T = \frac{\bar{\mathbf{a}}_M^{(1)}\mathbf{d}_M^{(12)H}}{\alpha_M^{f(1)}} + \frac{\mathbf{d}_M^{(2)}\bar{\mathbf{a}}_M^{(2)H}}{\alpha_M^{f(2)}} - \frac{\mathbf{Z}_M\bar{\mathbf{b}}_M^{(1)}\mathbf{c}_M^{(12)H}\mathbf{Z}_M^T}{\alpha_M^{b(1)}} - \frac{\mathbf{Z}_M\mathbf{c}_M^{(2)}\bar{\mathbf{b}}_M^{(2)H}\mathbf{Z}_M^T}{\alpha_M^{b(2)}} + \frac{\bar{\mathbf{w}}_M^{(1)}\bar{\mathbf{p}}_M^{(2)H}}{\alpha_M^{w(1)}} + \frac{\bar{\mathbf{p}}_M^{(1)}\bar{\mathbf{w}}_M^{(2)H}}{\alpha_M^{w(2)}} - \frac{\bar{\mathbf{v}}_M^{(1)}\bar{\mathbf{q}}_M^{(2)H}}{\alpha_M^{v(1)}} - \frac{\bar{\mathbf{q}}_M^{(1)}\bar{\mathbf{v}}_M^{(2)H}}{\alpha_M^{v(2)}}. \quad (72)$$

their evaluation on the unit circle, offering a a significant computational reduction compared to the direct (brute force) implementation. 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

Using the matrix inversion lemma (MIL) for partitioned matrices in (35) yields

$$[\mathbf{R}_M^{(i)}]^{-1} = \begin{bmatrix}[\mathbf{R}_{M-1}^{(i)}]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0\end{bmatrix} + \frac{\bar{\mathbf{b}}_M^{(i)}\bar{\mathbf{b}}_M^{(i)H}}{\alpha_M^{b(i)}} \quad (73)$$

$$= \begin{bmatrix}0 & \mathbf{0}^T \\ \mathbf{0} & [\widehat{\mathbf{R}}_{M-1}^{(i)}]^{-1}\end{bmatrix} + \frac{\bar{\mathbf{a}}_M^{(i)}\bar{\mathbf{a}}_M^{(i)H}}{\alpha_M^{f(i)}} \quad (74)$$

which, together with again using the MIL for rank-one modification on (41) and (42) implies

$$[\mathbf{R}_{M-1}^{(i)}]^{-1} = [\tilde{\mathbf{R}}_{M-1}^{(i)}]^{-1} - \frac{\mathbf{w}_{M-1}^{(i)}\mathbf{w}_{M-1}^{(i)H}}{\alpha_M^{w(i)}}, \quad (75)$$

$$\left[\tilde{\mathbf{R}}_{M-1}^{(i)}\right]^{-1} = [\widehat{\mathbf{R}}_{M-1}^{(i)}]^{-1} + \frac{\mathbf{v}_{M-1}^{(i)}\mathbf{v}_{M-1}^{(i)H}}{\alpha_m^{v(i)}}. \quad (76)$$

Furthermore, $\mathbf{R}_M^{(1,2)}$ is partitioned as

$$\mathbf{R}_M^{(1,2)} = \begin{bmatrix}\mathbf{R}_{M-1}^{(1,2)} & \mathbf{r}_{M-1}^{b(1,2)} \\ \mathbf{r}_{M-1}^{b(2,1)H} & r_M^{bo(1,2)}\end{bmatrix} = \begin{bmatrix}r_M^{fo(1,2)} & \mathbf{r}_{M-1}^{f(2,1)H} \\ \mathbf{r}_{M-1}^{f(1,2)} & \widehat{\mathbf{R}}_{M-1}^{(1,2)}\end{bmatrix}. \quad (77)$$

Combining (73), (74), and (77) yields

$$\mathbf{P}_M = \begin{bmatrix}\mathbf{P}_{M-1} & \mathbf{0} \\ \mathbf{0}^T & 0\end{bmatrix} + \frac{\bar{\mathbf{b}}_M^{(1)}\mathbf{c}_M^{(12)H}}{\alpha_M^{b(1)}} + \frac{\mathbf{c}_M^{(2)}\bar{\mathbf{b}}_M^{(2)H}}{\alpha_M^{b(2)}} \quad (78)$$

$$= \begin{bmatrix}0 & \mathbf{0}^T \\ \mathbf{0} & \widehat{\mathbf{P}}_{M-1}\end{bmatrix} + \frac{\bar{\mathbf{a}}_M^{(1)}\mathbf{d}_M^{(12)H}}{\alpha_M^{f(1)}} + \frac{\mathbf{d}_M^{(2)}\bar{\mathbf{a}}_M^{(2)H}}{\alpha_M^{f(2)}} \quad (79)$$

where

$$\mathbf{P}_{M-1} \triangleq [\mathbf{R}_{M-1}^{(i)}]^{-1}\mathbf{R}_{M-1}^{(12)}[\mathbf{R}_{M-1}^{(2)}]^{-1}, \quad (80)$$

$$\widehat{\mathbf{P}}_{M-1} \triangleq [\widehat{\mathbf{R}}_{M-1}^{(i)}]^{-1}\widehat{\mathbf{R}}_{M-1}^{(12)}[\widehat{\mathbf{R}}_{M-1}^{(2)}]^{-1} \quad (81)$$

with auxiliary variables being defined in Table I. It remains to show how the time shifted variables that are involved in (79) can be managed. A rank-one modification of (7) imply

$$\tilde{\mathbf{R}}_{M-1}^{(12)} = \mathbf{R}_{M-1}^{(12)} - \mathbf{x}_{M-1}^{(1)}(N-1)\mathbf{x}_{M-1}^{(2)H}(N-1) \quad (82)$$

$$\widehat{\mathbf{R}}_{M-1}^{(12)} = \tilde{\mathbf{R}}_{M-1}^{(12)} + \mathbf{x}_{M-1}^{(1)}(M-2)\mathbf{x}_{M-1}^{(2)H}(M-2) \quad (83)$$

which, combined with (75), (76), (82), and (83) yields

$$\mathbf{P}_{M-1} = \tilde{\mathbf{P}}_{M-1} - \frac{\mathbf{p}_{M-1}^{(1)}\mathbf{w}_{M-1}^{(2)H}}{\alpha_M^{w(2)}} - \frac{\mathbf{w}_{M-1}^{(1)}\mathbf{p}_{M-1}^{(2)H}}{\alpha_M^{w(1)}} \quad (84)$$

$$\tilde{\mathbf{P}}_{M-1} = \widehat{\mathbf{P}}_{M-1} + \frac{\mathbf{q}_{M-1}^{(1)}\mathbf{v}_{M-1}^{(2)H}}{\alpha_M^{v(2)}} + \frac{\mathbf{v}_{M-1}^{(1)}\mathbf{q}_{M-1}^{(2)H}}{\alpha_M^{v(1)}} \quad (85)$$

where

$$\tilde{\mathbf{P}}_{M-1} \triangleq [\tilde{\mathbf{R}}_{M-1}^{(1)}]^{-1}\tilde{\mathbf{R}}_{M-1}^{(12)}[\tilde{\mathbf{R}}_{M-1}^{(2)}]^{-1} \quad (86)$$

$$\hat{\mathbf{P}}_{M-1} \triangleq [\hat{\mathbf{R}}_{M-1}^{(1)}]^{-1}\hat{\mathbf{R}}_{M-1}^{(12)}[\hat{\mathbf{R}}_{M-1}^{(2)}]^{-1}. \quad (87)$$

Thus, using (78)-(85), one obtains the displacement of $\mathbf{P}_M$ as (72), given at the top of this page.

B. Displacement representation of $\mathbf{H}_{L,L}^{(12)}$

Consider the partitions of the data matrix (17), $\mathbf{X}_{M,L}^{(i)}$, as

$$\begin{bmatrix}\mathbf{X}_{M-1,L-1}^{(i)} & \mathbf{x}_{M-1}^{(i)}(N) \\ \times & \times\end{bmatrix} = \begin{bmatrix}\times & \times \\ \mathbf{x}_{M-1}^{(i)}(M-2) & \mathbf{X}_{M_1,L-1}^{(i)}\end{bmatrix}, \quad (90)$$

where the symbol $\times$ denotes unspecified terms. Combining (15), (78) and (90) results in

$$\mathbf{H}_{L,L}^{(12)} = \begin{bmatrix}\mathbf{H}_{L-1,L-1}^{(12)} & \times \\ \times & \times\end{bmatrix} + \frac{\mathbf{e}_L^{b(1)}\mathbf{e}_L^{c(12)H}}{\alpha_M^{b(1)}} + \frac{\mathbf{e}_L^{c(1)}\mathbf{e}_L^{b(2)H}}{\alpha_M^{b(2)}} \quad (91)$$

or, alternatively

$$\mathbf{H}_{L,L}^{(12)} = \begin{bmatrix}h_M^o & \mathbf{h}_{L-1}^{(21)H} \\ \mathbf{h}_{L-1}^{(12)} & \mathbf{H}_{L-1,L-1}^{(12)}\end{bmatrix} + \frac{\mathbf{e}_L^{a(1)}\mathbf{e}_L^{d(12)H}}{\alpha_M^{f(1)}} + \frac{\mathbf{e}_L^{d(1)}\mathbf{e}_L^{a(2)H}}{\alpha_M^{f(2)}} + \frac{\mathbf{e}_L^{w(1)}\mathbf{e}_L^{p(2)H}}{\alpha_M^{w(1)}} + \frac{\mathbf{e}_L^{p(1)}\mathbf{e}_L^{w(2)H}}{\alpha_M^{w(2)}} - \frac{\mathbf{e}_L^{v(1)}\mathbf{e}_L^{q(2)H}}{\alpha_M^{v(1)}} - \frac{\mathbf{e}_L^{q(1)}\mathbf{e}_L^{v(2)H}}{\alpha_M^{v(2)}} \quad (92)$$

where

$$h_M^o = \mathbf{x}_{M-1}^{(1)H}(M-2)\mathbf{P}_{M-1}\mathbf{x}_{M-1}^{(2)}(M-2) \quad (93)$$

$$\mathbf{h}_{L-1}^{(12)} = \mathbf{X}_{M-1,L-1}^{(1)H}\mathbf{P}_{M-1}\mathbf{x}_{M-1}^{(2)}(M-2) \quad (94)$$

$$\mathbf{h}_{L-1}^{(21)} = \mathbf{X}_{M-1,L-1}^{(2)H}\mathbf{P}_{M-1}^H\mathbf{x}_{M-1}^{(1)}(M-2) \quad (95)$$

Using (91) and (92), the displacement representation of $\mathbf{H}_{L,L}^{12}$ with respect to $\mathbf{Z}_L$ and $\mathbf{Z}_L^T$ takes the form of (88) that appears at the top of next page. It remains to show how (93)-(95) can be expressed in terms of variables already used by the algorithm. Using (78), (79), (84) and (85) together with (62)-(67) and taking into account the definition of the auxiliary variables presented in Table I, (93)-(95)

$$\mathbf{H}_{L,L}^{(12)} - \mathbf{Z}_L \mathbf{H}_{L,L}^{(12)} \mathbf{Z}_L^T = \begin{bmatrix} h_M^o & \mathbf{h}_{L-1}^{(21)H} \\ \mathbf{h}_{L-1}^{(12)} & \mathbf{0}_{L-1,L-1} \end{bmatrix} + \frac{\mathbf{e}_L^{a(1)}\mathbf{e}_L^{d(12)H}}{\alpha_M^{f(1)}} + \frac{\mathbf{e}_L^{d(1)}\mathbf{e}_L^{a(2)H}}{\alpha_M^{f(2)}} + \frac{\mathbf{e}_L^{w(1)}\mathbf{e}_L^{p(2)H}}{\alpha_M^{w(1)}} + \frac{\mathbf{e}_L^{p(1)}\mathbf{e}_L^{w(2)H}}{\alpha_M^{w(2)}} - \frac{\mathbf{e}_L^{v(1)}\mathbf{e}_L^{q(2)H}}{\alpha_M^{v(1)}} - \frac{\mathbf{e}_L^{q(1)}\mathbf{e}_L^{v(2)H}}{\alpha_M^{v(2)}} - \frac{\mathbf{Z}_L\mathbf{e}_L^{b(1)}\mathbf{e}_L^{c(12)H}\mathbf{Z}_L^T}{\alpha_M^{b(1)}} - \frac{\mathbf{Z}_L\mathbf{e}_L^{c(1)}\mathbf{e}_L^{b(2)H}\mathbf{Z}_L^T}{\alpha_M^{b(2)}} \tag{88}$$

$$\mathbf{H}_{L,L}^{(12)} - \mathbf{Z}_L \mathbf{H}_{L,L}^{(12)} \mathbf{Z}_L^T = \frac{\mathbf{e}_L^{a(1)}\mathbf{e}_L^{d(12)H}}{\alpha_M^{f(1)}} + \frac{\mathbf{e}_L^{d(1)}\mathbf{e}_L^{a(2)H}}{\alpha_M^{f(2)}} - \frac{\mathbf{Z}_L\mathbf{e}_L^{b(1)}\mathbf{e}_L^{c(12)H}\mathbf{Z}_L^T}{\alpha_M^{b(1)}} - \frac{\mathbf{Z}_L\mathbf{e}_L^{c(1)}\mathbf{e}_L^{b(2)H}\mathbf{Z}_L^T}{\alpha_M^{b(2)}} + \frac{\mathcal{P}[\mathbf{e}_L^{w(1)}]^{[0]}\left(\mathcal{P}[\mathbf{e}_L^{p(2)}]^{[0]}\right)^H}{\alpha_M^{w(1)}} + \frac{\mathcal{P}[\mathbf{e}_L^{p(1)}]^{[0]}\left(\mathcal{P}[\mathbf{e}_L^{w(2)}]^{[0]}\right)^H}{\alpha_M^{w(2)}} - \frac{\mathcal{P}[\mathbf{e}_L^{v(1)}]^{[\alpha_M^{v(1)}]}\left(\mathcal{P}[\mathbf{e}_L^{q(2)}]^{[1]}\right)^H}{\alpha_M^{v(1)}} - \frac{\mathcal{P}[\mathbf{e}_L^{q(1)}]^{[-q_M^o]}\left(\mathcal{P}[\mathbf{e}_L^{v(2)}]^{[\alpha_M^{v(2)}]}\right)^H}{\alpha_M^{v(2)}} + \mathbf{1}_L\mathbf{1}_L^T \tag{89}$$

take the form

$$h_M^o = q_M^0 + \frac{\mathbf{e}_{L,1}^{q(1)}\mathbf{e}_{L,1}^{v(2)*}}{\alpha_M^{v(2)}} + \frac{\mathbf{e}_{L,1}^{v(1)}\mathbf{e}_{L,1}^{q(2)*}}{\alpha_M^{v(1)}} - \frac{\mathbf{e}_{L,1}^{p(1)}\mathbf{e}_{L,1}^{w(2)*}}{\alpha_M^{w(2)}} - \frac{\mathbf{e}_{L,1}^{w(1)}\mathbf{e}_{L,1}^{p(2)*}}{\alpha_M^{w(1)}} \tag{96}$$

$$\mathbf{h}_{L-1}^{(12)} = -\frac{\mathbf{e}_{L,2:L}^{q(1)}}{\alpha_M^{v(2)}} + \frac{\mathbf{e}_{L,2:L}^{v(1)}\left(-1 + \mathbf{e}_{L,1}^{q(2)*}\right)}{\alpha_M^{v(1)}} - \frac{\mathbf{e}_{L,2:L}^{p(1)}\mathbf{e}_{L,1}^{w(1)*}}{\alpha_M^{w(2)}} - \frac{\mathbf{e}_{L,2:L}^{w(1)}\mathbf{e}_{L,1}^{p(2)*}}{\alpha_M^{w(1)}} \tag{97}$$

$$\mathbf{h}_{L-1}^{(21)} = -\frac{\mathbf{e}_{L,2:L}^{q(2)}}{\alpha_M^{v(1)}} + \frac{\mathbf{e}_{L,2:L}^{v(2)}\left(q_M^{o*} + \mathbf{e}_{L,1}^{q(1)*}\right)}{\alpha_M^{v(2)}} - \frac{\mathbf{e}_{L,2:L}^{w(1)}\mathbf{e}_{L,1}^{p(1)*}}{\alpha_M^{w(2)}} - \frac{\mathbf{e}_{L,2:L}^{p(2)}\mathbf{e}_{L,1}^{w(1)*}}{\alpha_M^{w(1)}} \tag{98}$$

with $\mathbf{x}_{L,1}$ and $\mathbf{x}_{L,2:L}$ denoting the first element, and all but the first elements of the vector $\mathbf{x}_L$, respectively, i.e., $\mathbf{x}_L = [\mathbf{x}_{L,1}\ \mathbf{x}_{L,2:L}^T]^T$. Finally, using (96)-(98) and (88) one gets the displacement of $\mathbf{H}_{L,L}^{(12)}$ as it appears in (89) at the top of this page.

REFERENCES

[1] J. Benesty, J. Chen, and Y. Huang, "A Generalized MVDR Spectrum," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 827–830, December 2005.

[2] A. Jakobsson, S. R. Alty, and J. Benesty, "Estimating and Time-Updating the 2-D Coherence Spectrum," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2350–2354, May 2007.

[3] N. R. Butt and A. Jakobsson, "Coherence Spectrum Estimation From Nonuniformly Sampled Sequences," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 339–342, April 2010.

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

[5] C. Zheng, Y. Zhou, and X. Li, "Generalised framework for nonparametric coherence function estimation," *Electronics Letters*, vol. 46, no. 6, pp. 450–452, 18 2010.

[6] M. M. Hyder and K. Mahata, "Coherent Spectral Analysis of Asynchronously Sampled Signals," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 126–129, Feb. 2011.

[7] G.-O. Glentis, "A Fast Algorithm for APES and Capon Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4207–4220, Sept. 2008.

[8] G.-O. Glentis and A. Jakobsson, "Efficient Implementation of Iterative Adaptive Approach Spectral Estimation Techniques," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4154–4167, Sept. 2011.

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

[10] P. Stoica, A. Jakobsson, and J. Li, "Matched-Filterbank Interpretation of Some Spectral Estimators," *Signal Processing*, vol. 66, no. 1, pp. 45–59, April 1998.

[11] G. O. Glentis, "Efficient Algorithms for Adaptive Capon and APES Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 84–96, Jan. 2010.

[12] T. Kailath and A. H. Sayed, "Displacement Structure: Theory and Applications," *SIAM Review*, vol. 37, no. 3, pp. 297–386, September 1995.

[13] I. Gohberg and V. Olshevksy, "Complexity of multiplication with vectors for structured matrices," *Linear Algebra Appl.*, vol. 202, pp. 163–192, 1994.

[14] D. Wood, "Product rules for the displacement of near-Toeplitz matrices," *Linear Algebra Appl.*, vol. 188/189, pp. 641–663, 1993.

[15] T. Kailath and A. H. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, USA, 1999.