Signal Processing 92 (2012) 617–624

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

On Jacobi-type methods for blind equalization of paraunitary channels

Mikael Sørensen [a,*], Lieven De Lathauwer [a], Sylvie Icart [b], Luc Deneire [b]

[a] K.U.Leuven - E.E. Dept. (ESAT) - SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
[b] Les Algorithmes - Euclide-B, 06903 Sophia Antipolis, France

ARTICLE INFO

Article history:
Received 20 September 2010
Received in revised form
31 March 2011
Accepted 6 July 2011
Available online 19 July 2011

Keywords:
Blind equalization
Blind deconvolution
Tensors
Jacobi method

ABSTRACT

In this paper a study of the cumulant-based blind equalization algorithms PAJOD and PAFA is conducted. Both algorithms assume that the data have been pre-whitened and hence the problem reduces to the estimation of paraunitary channels. The main contribution of this paper is an efficient implementation of the PAJOD algorithm called PAJOD2. Second, a performance comparison between the PAJOD, PAJOD2 and PAFA algorithms is reported.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Blind equalization of linear time-invariant MIMO channels refers to channel equalization techniques where only the observed signal is known. The observed signal is assumed to consist of an unknown convolutive mixture of input signals. The cumulant-based blind equalization algorithms Partial Approximate JOint Diagonalization (PAJOD) and PAraunitary FActorization (PAFA) were proposed in [1] and in [2], respectively. Both are based on contrast maximization and the working assumption for both algorithms is that the data have been pre-whitened. A method to perform pre-whitening has been proposed in [3].

Due to the pre-whitening the problem reduces to a search for a paraunitary equalizer. The PAFA algorithm looks for a paraunitary equalizer while the PAJOD algorithm only searches for a semi-unitary equalizer. The PAFA and PAJOD algorithms both consist of a Jacobi-type iteration where the Jacobi subproblem is solved by a computationally demanding resultant-based procedure. The PAFA algorithm requires the rooting of a 56th degree polynomial in each Jacobi subproblem. PAJOD requires the rooting of either a 3rd or 24th degree polynomial in each of its Jacobi subproblems, as will be explained later.

The main contribution of this paper is a more efficient implementation of the PAJOD algorithm called PAJOD2. Part of this work has been presented in [4]. The paraunitary equalizer PAFA fully takes the structure of the problem into account while the semi-unitary equalizer PAJOD only partially exploits the structure of the problem. However, the PAJOD algorithm is less computationally demanding than the PAFA algorithm. Hence, a comparison of the PAJOD, PAJOD2 and PAFA algorithms based on computer simulations will be reported.

The paper is organized as follows. First the notation used throughout the paper will be introduced. Since the algorithms are based on paraunitary filters and contrast optimization, a few basic notions about paraunitary filters and contrasts will be presented before discussing the system model. Next, in Section 2 a brief review of the PAJOD and PAFA algorithms is given. Furthermore, the more efficient implementation of the PAJOD algorithm called PAJOD2 will be presented. Section 3 will compare the PAJOD, PAJOD2 and PAFA methods based on computer

* Corresponding author.
E-mail addresses: Mikael.Sorensen@kuleuven-kortrijk.be (M. Sørensen), Lieven.DeLathauwer@kuleuven-kortrijk.be (L. De Lathauwer), icart@i3s.unice.fr (S. Icart), deneire@i3s.unice.fr (L. Deneire).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.07.005

simulations. We end the paper with a conclusion in Section 4.

1.1. Notation

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{C}[z]$ denote the set of natural, integer, real, complex numbers and the set of polynomials in z with coefficients in $\mathbb{C}$, respectively. Furthermore, let $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^\dagger$, $\mathrm{Re}\{\cdot\}$, $\mathrm{Im}\{\cdot\}$ and $\|\cdot\|_F$ denote the conjugate, transpose, conjugate-transpose, pseudo-inverse, real part, imaginary part and the Frobenius norm of a matrix, respectively. The operator $\mathrm{diag}(\cdot)$ sets all the off-diagonal elements of a matrix equal to zero. Let $\mathbf{I}_R \in \mathbb{C}^{R\times R}$ denote the identity matrix. Given $\mathbf{A} \in \mathbb{C}^{m\times n}$, then $\mathbf{A}_{ij}$ denotes the ith row–jth column entry of $\mathbf{A}$. Finally, let $\mathbf{H}(z) = \sum_n \mathbf{H}(n)z^{-n}$.

1.2. Paraunitary filter

A filter $\mathbf{H}(z) = \sum_{l=0}^{L-1} \mathbf{H}(l)z^{-l} \in \mathbb{C}[z]^{R\times R}$ is paraunitary if $\mathbf{H}^H(1/z^*)\mathbf{H}(z) = \mathbf{I}_R$. The paraunitary filter $\mathbf{H}(z)$ satisfies the properties [5]:

- The inverse filter $\mathbf{H}^{-1}(z) = \mathbf{H}^H(1/z^*)$ is paraunitary.
- The channel impulse response matrix $\overline{\mathbf{H}} = [\mathbf{H}(0), \mathbf{H}(1), \ldots, \mathbf{H}(L-1)] \in \mathbb{C}^{R\times RL}$ is a semi-unitary matrix, i.e., $\overline{\mathbf{H}}\overline{\mathbf{H}}^H = \mathbf{I}_R$.

1.3. Contrast optimization

The notion of contrast optimization was introduced in [6] and applied in the framework of MIMO equalization in [7]. Let $\mathcal{H}$ and $\mathcal{S}$ be the set of paraunitary filters and the set of transmitted symbol sequences, respectively. Furthermore, let $\mathcal{H}\cdot\mathcal{S}$ denote the set of recovered symbol sequences and $\mathcal{T}$ denote the set of paraunitary equalizers that do not violate the working assumptions on $\mathcal{S}$ specified below. Moreover, let $\mathbf{I}$ denote the identity operator, then a function $\mathcal{J}(\mathbf{H};\mathbf{x})$ is called a contrast if it satisfies the properties [7]:

- *Invariance*: $\mathcal{J}(\mathbf{H};\mathbf{x}) = \mathcal{J}(\mathbf{I};\mathbf{x}), \forall \mathbf{H} \in \mathcal{T}, \forall \mathbf{x} \in \mathcal{H}\cdot\mathcal{S}$.
- *Domination*: $\mathcal{J}(\mathbf{H};\mathbf{x}) \le \mathcal{J}(\mathbf{I};\mathbf{x}), \forall \mathbf{H} \in \mathcal{H}, \forall \mathbf{x} \in \mathcal{S}$.
- *Discrimination*: $\mathcal{J}(\mathbf{H};\mathbf{x}) = \mathcal{J}(\mathbf{I};\mathbf{x}), \forall \mathbf{x} \in \mathcal{S} \Rightarrow \mathbf{H} \in \mathcal{T}$.

Under the assumption that there exists an equalizer that will fully recover the symbols, an equalizer corresponding to the global maximum of the contrast function is guaranteed to recover the symbol sequence.

1.3.1. System model

Let $\mathbf{s}(n)$, $\mathbf{x}(n) \in \mathbb{C}^R$ be the symbol and observation vector at time instant $n \in \mathbb{N}$, respectively. Assume that $\mathbf{s}(n)$ and $\mathbf{x}(n)$ are related via

$$\mathbf{x}(n) = \sum_{k=0}^{K-1} \mathbf{F}(k)\mathbf{s}(n-k),$$

where $\mathbf{F}(k) \in \mathbb{C}^{R\times R}$, $k \in [0,K-1]$, is the channel impulse response of the paraunitary filter $\mathbf{F}(z)$. The problem is to estimate the symbol sequence $\{\mathbf{s}(n)\}$ from the observation sequence $\{\mathbf{x}(n)\}$. This is done by the equalizer $\mathbf{H}(z)$ such that

$$\mathbf{y}(n) = \sum_{l=0}^{L-1} \mathbf{H}(l)\mathbf{x}(n-l) = \sum_{l=0}^{L-1}\sum_{k=0}^{K-1} \mathbf{H}(l)\mathbf{F}(k)\mathbf{s}(n-l-k),$$

where $\mathbf{H}(l) \in \mathbb{C}^{R\times R}$, $l \in [0,L-1]$, are the channel impulse response coefficients of $\mathbf{H}(z)$ and $\mathbf{y}(n)$ is the recovered symbol vector at time instant n. Since $\mathbf{F}(z)$ is paraunitary, we know that the equalizer $\mathbf{H}(z)$ is also paraunitary and that $K=L$. In the PAJOD and PAFA methods, the working assumptions on the transmitted signal sequences are:

- $\mathbf{s}_r(n)$ are mutually independent i.i.d., zero-mean processes with unit-variance for all $r \in [1,R]$.
- $\mathbf{s}(n)$ is stationary up to order 4 and hence the marginal cumulants of order 4 do not depend on n.
- At most one source has zero marginal cumulant of order 4.

2. Paraunitary equalization algorithms

The PAJOD and PAFA blind paraunitary equalization algorithms will be briefly reviewed and the computationally improved version of PAJOD algorithm called PAJOD2 will be introduced.

2.1. PAJOD

In [1], the cumulants of the observed data are stored in a set of $RL\times RL$ matrices $\mathbf{M}(\mathbf{b},\gamma)$ in such a way that for a fixed pair $(\mathbf{b},\gamma) = ([b_1,b_2],[\gamma_1,\gamma_2])$ we have the relation

$$\mathbf{M}_{\alpha_1 R+a_1,\alpha_2 R+a_2}(\mathbf{b},\gamma) = \mathrm{Cum}[y_{a_1}(n-\alpha_1), y^*_{a_2}(n-\alpha_2), y_{b_1}(n-\gamma_1), y^*_{b_2}(n-\gamma_2)].$$

Moreover, in [1] it is shown that the function

$$\mathcal{J}_2^2 = \sum_{\mathbf{b},\gamma} \|\mathrm{diag}(\overline{\mathbf{H}}\mathbf{M}(\mathbf{b},\gamma)\overline{\mathbf{H}}^H)\|_F^2 \tag{1}$$

is a contrast function, where $\|\mathrm{diag}(\mathbf{A})\|_F^2 = \sum_i |\mathbf{A}_{ii}|^2$ and $\overline{\mathbf{H}} = [\mathbf{H}(0),\mathbf{H}(1),\ldots,\mathbf{H}(L-1)] \in \mathbb{C}^{R\times RL}$. Due to the paraunitary assumption on $\mathbf{H}(z)$, the matrix $\overline{\mathbf{H}}$ is a semi-unitary matrix, i.e., $\overline{\mathbf{H}}\overline{\mathbf{H}}^H = \mathbf{I}_R$.

2.1.1. Jacobi procedure for semi-unitary matrices

To numerically find the semi-unitary matrix $\overline{\mathbf{H}}$ that will maximize the contrast (1) a Jacobi procedure was proposed in [1]. This procedure can be seen as a double extension of the JADE algorithm [8,9]. First, the unknown matrix is semi-unitary instead of unitary. Second, only the R first diagonal entries are of interest.

A Jacobi procedure is based on the fact that any $RL\times RL$ unitary matrix with determinant equal to one can be parametrized as a product of Givens rotations [10]:

$$\mathbf{V} = \prod_{p=1}^{RL-1}\prod_{q=p+1}^{RL} \Theta[p,q]^H,$$

where $\Theta[p,q]$ is equal to the identity matrix, except for entries

$$\Theta_{pp}[p,q] = \Theta_{qq}[p,q] = \cos(\theta[p,q]),$$

$$\Theta_{qp}[p,q] = -\Theta_{pq}[p,q]^* = \sin(\theta[p,q])e^{i\phi[p,q]}, \theta[p,q], \phi[p,q] \in \mathbb{R}.$$

Let $\mathbf{V}$ denote the product of Givens matrices with the initial value $\mathbf{V}=\mathbf{I}_{RL}$. The updating rules are given by $\mathbf{M}(\mathbf{b},\gamma)\leftarrow\Theta[p,q]^H\mathbf{M}(\mathbf{b},\gamma)\Theta[p,q]$ and $\mathbf{V}\leftarrow\Theta[p,q]^H\mathbf{V}$. In the PAJOD algorithm the semi-unitary matrix $\overline{\mathbf{H}}$ is determined as the first R rows of the unitary matrix $\mathbf{V}$. The Givens rotation matrix $\Theta[p,q]$ is chosen as the maximizer of

$$\begin{aligned}\mathcal{J}_2^2(p,q)&=\sum_{\mathbf{b},\gamma}\sum_{k=1}^{R}|(\Theta[p,q]^H\mathbf{M}(\mathbf{b},\gamma)\Theta[p,q])_{kk}|^2\\&=\sum_{\mathbf{b},\gamma}\sum_{k=1}^{R}\left|\sum_{\eta,\mu=1}^{RL}\Theta^*_{\eta k}[p,q]\Theta_{\mu k}[p,q]\mathbf{M}(\mathbf{b},\gamma)_{\eta\mu}\right|^2\\&=\sum_{\omega}\sum_{k=1}^{R}\left|\sum_{\eta,\mu=1}^{RL}\Theta^*_{\eta k}[p,q]\Theta_{\mu k}[p,q]\mathbf{M}^{(\omega)}_{\eta\mu}\right|^2.\end{aligned}\tag{2}$$

Notice that we denote $\mathbf{M}(\mathbf{b},\gamma)$ by $\mathbf{M}^{(\omega)}$ for simplicity. Since plane rotations where $p>R$ do not have any effect on the first R rows of the matrices $\mathbf{M}^{(\omega)}$ only Givens rotations where $p\le R$ are considered. Furthermore one has to distinguish between the cases where $q\le R$ and $q>R$. The problems are illustrated in Fig. 1. When $q\le R$, then the aim of the Givens rotation matrix $\Theta[p,q]$ is to jointly diagonalize the set of matrices $\{\mathbf{M}^{(\omega)}\}$ by maximizing the entries $\mathbf{M}^{(\omega)}_{pp}$ and $\mathbf{M}^{(\omega)}_{qq}$ for all ω as illustrated in Fig. 1(a). On the other hand, when $q>R$, then the aim of the Givens rotation matrix $\Theta[p,q]$ reduces to jointly diagonalize the set of matrices $\{\mathbf{M}^{(\omega)}\}$ by maximizing the entry $\mathbf{M}^{(\omega)}_{pp}$ for all ω as illustrated in Fig. 1(b).

Let $\overline{\mathbf{M}}^{(\omega)}=\Theta[p,q]^H\mathbf{M}^{(\omega)}\Theta[p,q]$ and for notational convenience let $c=\cos(\theta[p,q])$ and $s=\sin(\theta[p,q])e^{i\phi[p,q]}$. Then for the case where $q\le R$ Eq. (2) is equal to

$$\mathcal{J}_2^2(p,q)|_{q\le R}=\sum_{\omega}|\overline{\mathbf{M}}^{(\omega)}_{pp}|^2+|\overline{\mathbf{M}}^{(\omega)}_{qq}|^2,\tag{3}$$

where

$$\overline{\mathbf{M}}^{(\omega)}_{pp}=[c\ \ s^*]\begin{bmatrix}\mathbf{M}^{(\omega)}_{pp}&\mathbf{M}^{(\omega)}_{pq}\\\mathbf{M}^{(\omega)}_{qp}&\mathbf{M}^{(\omega)}_{qq}\end{bmatrix}\begin{bmatrix}c\\s\end{bmatrix}$$

and

$$\overline{\mathbf{M}}^{(\omega)}_{qq}=[c\ \ -s]\begin{bmatrix}\mathbf{M}^{(\omega)}_{pp}&\mathbf{M}^{(\omega)}_{pq}\\\mathbf{M}^{(\omega)}_{qp}&\mathbf{M}^{(\omega)}_{qq}\end{bmatrix}\begin{bmatrix}c\\-s^*\end{bmatrix}.$$

The maximization problem (3) is equivalent to the JADE diagonalization problem and therefore the JADE algorithm [8] can be applied to solve this problem.

When $q>R$ only the first diagonal term should be maximized, and Eq. (2) reduces to

$$\mathcal{J}_2^2(p,q)|_{q>R}=\sum_{p}|\overline{\mathbf{M}}^{(\omega)}_{pp}|^2.\tag{4}$$

In [1] a resultant based [11] approach was taken to solve the maximization problem (4). It amounted to the rooting of a polynomial of degree 24 that has at most eight real roots.

An outline of the PAJOD algorithm is given in Algorithm 1.

Algorithm 1. Outline of the PAJOD procedure.

```
Estimate the cumulant tensor M(b,γ)
Initialize 𝕍 = I_RL
Repeat until convergence
for p=1 to R do
  for q=p+1 to RL do
    if q ≤ R then
      calculate optimal Θ[p,q] from J_2^2(p,q)|_{q≤R} according to [8].
    else
      calculate optimal Θ[p,q] from J_2^2(p,q)|_{q>R} according to [1].
    end if
    M^(k) ← Θ[p,q]^H M^(k) Θ[p,q]
    𝕍 ← Θ[p,q]^H 𝕍
  end for
end for
Set ℍ = 𝕍(:,1:R)
```

2.2. PAJOD2

This section will introduce the PAJOD2 algorithm, which is a computationally improved version of the PAJOD algorithm. The overall algorithmic structure is still as in Algorithm 1. When $q\le R$, then the PAJOD2 algorithm will apply the JADE algorithm to solve the Jacobi subproblem, just as in the PAJOD case.

However, for the case $q>R$ a more efficient eigenvector-based approach will be proposed. In the derivation we

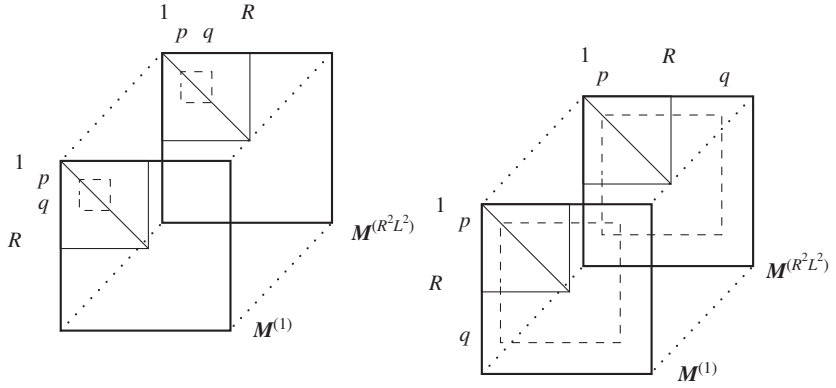


Fig. 1. The figures illustrate the PAJOD optimization problems for the case when $\mathbf{b}$ varies in $\{1,\ldots,R\}^2$ and γ in $\{0,\ldots,L-1\}^2$. (a) $q\le R$. (b) $q>R$.

will make use of the trigonometric identities

$$2\cos^2(\theta) = (1+\cos(2\theta)),$$

$$2\sin^2(\theta) = (1-\cos(2\theta)),$$

$$2\cos(\theta)\sin(\theta) = \sin(2\theta),$$

$$\cos(2\phi) = \cos^2(\phi) - \sin^2(\phi),$$

$$\sin(2\phi) = 2\cos(\phi)\sin(\phi).$$

Let $\theta = \theta[p,q]$, $\phi = \phi[p,q]$, $\hat{c} = \cos(2\theta)$, $\hat{s} = \sin(2\theta)$, $\alpha^{(\omega)} = \mathbf{M}_{pp}^{(\omega)} - \mathbf{M}_{qq}^{(\omega)}$ and $\beta^{(\omega)} = \mathbf{M}_{pp}^{(\omega)} + \mathbf{M}_{qq}^{(\omega)}$, then Eq. (4) can be written as

$$\begin{aligned}\mathcal{J}_2^2(p,q)|_{q>R} = &\frac{1}{4}\sum_{\omega}|\beta^{(\omega)}|^2 + |\alpha^{(\omega)}|^2\hat{c}^2 + 2\,\mathrm{Re}\{\beta^{(\omega)*}\alpha^{(\omega)}\}\hat{c} \\ &+ (|\mathbf{M}_{pq}^{(\omega)}|^2 + |\mathbf{M}_{qp}^{(\omega)}|^2 + 2\mathrm{Re}\{\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}e^{i2\phi}\})\hat{s}^2 \\ &+ 2\,\mathrm{Re}\{\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)*}e^{i\phi} + \mathbf{M}_{qp}^{(\omega)*}e^{-i\phi})\}\hat{c}\hat{s} \\ &- 2\,\mathrm{Re}\{\beta^{(\omega)*}(\mathbf{M}_{pq}^{(\omega)}e^{-i\phi} + \mathbf{M}_{qp}^{(\omega)}e^{i\phi})\}\hat{s}. \end{aligned} \tag{5}$$

By inspection of (5) we can identify the term independent of θ and ϕ as

$$k = \frac{1}{4}\sum_{\omega}|\beta^{(\omega)}|^2 = \frac{1}{4}\sum_{\omega}|\mathbf{M}_{pp}^{(\omega)} + \mathbf{M}_{qq}^{(\omega)}|^2. \tag{6}$$

The linear terms in the variables $\hat{c}$ and $\hat{s}$ of (5) can be written as

$$\begin{aligned} l(\hat{c},\hat{s}) &= \frac{1}{2}\sum_{\omega}\mathrm{Re}\{\beta^{(\omega)*}\alpha^{(\omega)}\}\hat{c} - \mathrm{Re}\{\beta^{(\omega)*}(\mathbf{M}_{pq}^{(\omega)}e^{-i\phi} + \mathbf{M}_{qp}^{(\omega)}e^{i\phi})\}\hat{s} \\ &= \frac{1}{2}\sum_{\omega}\mathrm{Re}\{\beta^{(\omega)*}\alpha^{(\omega)}\}\hat{c} - \mathrm{Re}\{\beta^{(\omega)*}(\mathbf{M}_{pq}^{(\omega)} + \mathbf{M}_{qp}^{(\omega)})\}\hat{s}\cos(\phi) \\ &\quad + \mathrm{Re}\{i\beta^{(\omega)*}(\mathbf{M}_{pq}^{(\omega)} - \mathbf{M}_{qp}^{(\omega)})\}\hat{s}\,\sin(\phi) \\ &= \sum_{\omega}\mathbf{g}^{(\omega)T}\mathbf{v}, \end{aligned} \tag{7}$$

where

$$\mathbf{v} = \begin{bmatrix} \cos(2\theta[p,q]) \\ \sin(2\theta[p,q])\cos(\phi[p,q]) \\ \sin(2\theta[p,q])\sin(\phi[p,q]) \end{bmatrix},$$

$$\mathbf{z}^{(\omega)} = \frac{1}{2}\begin{bmatrix} \mathbf{M}_{pp}^{(\omega)} - \mathbf{M}_{qq}^{(\omega)} \\ -(\mathbf{M}_{pq}^{(\omega)} + \mathbf{M}_{qp}^{(\omega)}) \\ i(\mathbf{M}_{pq}^{(\omega)} - \mathbf{M}_{qp}^{(\omega)}) \end{bmatrix},$$

$$\mathbf{g}^{(\omega)} = \mathrm{Re}\{(\mathbf{M}_{pp}^{(\omega)} + \mathbf{M}_{qq}^{(\omega)})^*\mathbf{z}^{(\omega)}\}.$$

The quadratic terms in the variables $\hat{c}$ and $\hat{s}$ of (5) will now be written as

$$\begin{aligned} q(\hat{c},\hat{s}) &= \frac{1}{4}\sum_{\omega}|\alpha^{(\omega)}|^2\hat{c}^2 + 2\,\mathrm{Re}\{\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)*}e^{i\phi} + \mathbf{M}_{qp}^{(\omega)*}e^{-i\phi})\}\hat{c}\hat{s} \\ &\quad + (|\mathbf{M}_{pq}^{(\omega)}|^2 + |\mathbf{M}_{qp}^{(\omega)}|^2 + 2\,\mathrm{Re}\{\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}e^{i2\phi}\})\hat{s}^2 \\ &= \frac{1}{4}\sum_{\omega}|\alpha^{(\omega)}|^2\hat{c}^2 + (|\mathbf{M}_{pq}^{(\omega)}|^2 + |\mathbf{M}_{qp}^{(\omega)}|^2)\hat{s}^2(\cos^2(\phi) \\ &\quad + \sin^2(\phi)) + 2\,\mathrm{Re}\{\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}\}\hat{s}^2\cos(2\phi) \\ &\quad + 2\,\mathrm{Re}\{i\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}\}\hat{s}^2\sin(2\phi) \\ &\quad + 2\,\mathrm{Re}\{\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)} + \mathbf{M}_{qp}^{(\omega)})^*\}\hat{c}\hat{s}\,\cos(\phi) \\ &\quad + 2\,\mathrm{Re}\{i\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)} - \mathbf{M}_{qp}^{(\omega)})^*\}\hat{c}\hat{s}\,\sin(\phi) \\ &= \frac{1}{4}\sum_{\omega}|\alpha^{(\omega)}|^2\hat{c}^2 + (|\mathbf{M}_{pq}^{(\omega)}|^2 + |\mathbf{M}_{qp}^{(\omega)}|^2)\hat{s}^2(\cos^2(\phi) \\ &\quad + \sin^2(\phi)) + 2\,\mathrm{Re}\{\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}\}\hat{s}^2(\cos^2(\phi) - \sin^2(\phi)) \\ &\quad + 4\,\mathrm{Re}\{i\mathbf{M}_{pq}^{(\omega)*}\mathbf{M}_{qp}^{(\omega)}\}\hat{s}^2\cos(\phi)\sin(\phi) \\ &\quad + 2\,\mathrm{Re}\{\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)} + \mathbf{M}_{qp}^{(\omega)})^*\}\hat{c}\hat{s}\cos(\phi) \\ &\quad + 2\,\mathrm{Re}\{\alpha^{(\omega)}(\mathbf{M}_{pq}^{(\omega)} - \mathbf{M}_{qp}^{(\omega)})^*\}\hat{c}\hat{s}\sin(\phi) \\ &= \mathbf{v}^T\sum_{\omega}\mathbf{G}^{(\omega)}\mathbf{v}, \end{aligned} \tag{8}$$

where $\mathbf{G}^{(\omega)} = \mathrm{Re}\{\mathbf{z}^{(\omega)}\mathbf{z}^{(\omega)H}\}$.

From Eqs. (6)–(8), Eq. (4) can be reformulated as

$$\mathcal{J}_2^2(p,q)|_{q>R} = \mathbf{v}^T\mathbf{G}\mathbf{v} + \mathbf{g}^T\mathbf{v} + k, \tag{9}$$

where $\mathbf{G} = \sum_{\omega}\mathbf{G}^{(\omega)}$ and $\mathbf{g} = \sum_{\omega}\mathbf{g}^{(\omega)}$. We should maximize (9) under the constraint $\|\mathbf{v}\|_F^2 = 1$.

Maximizing (9) subject to the constraint that $\|\mathbf{v}\|_F^2 = 1$ is a classical problem. In the context of source separation, it for instance appeared in [12]. Using the Lagrange multiplier method leads to

$$2(\mathbf{G} + \lambda\mathbf{I}_3)\mathbf{v} + \mathbf{g} = 0, \quad \lambda \in \mathbb{R}. \tag{10}$$

Assuming that $(\mathbf{G} + \lambda\mathbf{I}_3)^{-1}$ exists, we have

$$\mathbf{v} = -\tfrac{1}{2}(\mathbf{G} + \lambda\mathbf{I}_3)^{-1}\mathbf{g}.$$

Given the eigenvalue decomposition $\mathbf{G} = \mathbf{E}\Lambda\mathbf{E}^T$, we have

$$\|\mathbf{v}\|_F^2 = 1 \Leftrightarrow \frac{1}{4}\sum_{n=1}^{3}\frac{(\mathbf{E}_n^T\mathbf{g})^2}{(\Lambda_{nn} + \lambda)^2} = 1, \tag{11}$$

where $\mathbf{E}_n$ and Λ_{nn} denote the nth eigenvector and eigenvalue of $\mathbf{G}$, respectively. From (11) one can deduce that the problem amounts to rooting a polynomial of degree 6 and thereafter selecting the root of which the corresponding $\mathbf{v}$ maximizes $\mathcal{J}_2^2(p,q)$.

If $(\mathbf{G} + \lambda\mathbf{I}_3)^{-1}$ does not exist,[1] which could occur if $\lambda = 0$ and $\mathbf{G}$ is singular or when $\lambda = -\Lambda_{nn}$ for some n, then we have to resort to (10) for the computation of $\mathbf{v}$ [12]:

$$\mathbf{v} = -\tfrac{1}{2}(\mathbf{G} - \Lambda_{nn}\mathbf{I}_3)^{\dagger}\mathbf{g} + c_n\mathbf{E}_n,$$

where c_n is a real constant chosen such that $\|\mathbf{v}\|_F^2 = 1$ and $\mathcal{J}_2^2(p,q)$ is maximum. If it exists, it is given by

$$c_n = \mathrm{sign}(\mathbf{E}_n^T\mathbf{g})\sqrt{1 - \|(\mathbf{G} - \Lambda_{nn}\mathbf{I}_3)^{\dagger}\mathbf{g}\|^2/4}.$$

We emphasize that PAJOD and PAJOD2 compute the same filter. However, the steps taken when $p > q$ are different. An outline of the PAJOD2 algorithm is given in Algorithm 2. By comparing Algorithms 1 and 2 it is clear that the difference between the PAJOD and PAJOD2 algorithms is in the computation of the optimal $\Theta[p,q]$ from $\mathcal{J}_2^2(p,q)|_{q>R}$.

Algorithm 2. Outline of the PAJOD2 procedure.

Estimate the cumulant tensor $\mathcal{M}(\mathbf{b},\gamma)$
Initialize $\mathbb{V} = \mathbf{I}_{RL}$
Repeat until convergence
for $p = 1$ to R **do**

[1] This case has not been observed in our simulations.

for $q=p+1$ to RL **do**
  if $q \le R$ **then**
    calculate optimal $\Theta[p,q]$ from $\mathcal{J}_2^2(p,q)|_{q \le R}$ according to [8].
  else
    calculate optimal $\Theta[p,q]$ from $\mathcal{J}_2^2(p,q)|_{q > R}$ according to Section 2.2.
  end if
  $\mathbf{M}^{(k)} \leftarrow \Theta[p,q]^H \mathbf{M}^{(k)} \Theta[p,q]$
  $\mathbb{V} \leftarrow \Theta[p,q]^H \mathbb{V}$
end for
end for
Set $\mathbb{H} = \mathbb{V}(:,1:R)$

2.3. PAFA

The PAFA algorithm is based on the fact that any FIR paraunitary filter $\mathbf{H}(z) \in \mathbb{C}[z]^{R\times R}$ of length L can be factorized as [5]

$$\mathbf{H}(z) = \mathbf{Q}^{(M)}\mathbf{Z}(z)\mathbf{Q}^{(M-1)}\mathbf{Z}(z)\cdots\mathbf{Z}(z)\mathbf{Q}^{(0)}, \tag{12}$$

where M is the McMillan degree of $\mathbf{H}(z)$, $\mathbf{Q} \in \mathbb{C}^{R\times R}$ is unitary and $\mathbf{Z}(z) = \left[\begin{smallmatrix} \mathbf{I}_{R-1} & \mathbf{0} \\ \mathbf{0} & z^{-1} \end{smallmatrix}\right]$.

The factorization (12) fully takes the paraunitary structure of the problem into account while PAJOD does not. An approach that cyclically estimates one unitary matrix $\mathbf{Q}^{(m)}$ while the other matrices are fixed was proposed in [2].

We now describe how unitary matrix $\mathbf{Q}^{(m)}$ is estimated while the other matrices are fixed. First (12) is reformulated as

$$\mathbf{H}(z) = \mathbf{A}(z)\mathbf{Q}^{(m)}\mathbf{B}(z), \tag{13}$$

where $\mathbf{A}(z) = \mathbf{Q}^{(M)}\mathbf{Z}(z)\cdots\mathbf{Z}(z)\mathbf{Q}^{(m+1)}$ and $\mathbf{B}(z) = \mathbf{Q}^{(m-1)}\mathbf{Z}(z)\cdots\mathbf{Z}(z)\mathbf{Q}^{(0)}$ are of length $l_A \le L$ and $l_B \le L$, respectively, which satisfy $l_A + l_B = L$. The relation between the signal vector after the equalization of $\mathbf{y}(n)$ and the observation vector $\mathbf{x}(n)$ can now be written as

$$\mathbf{y}_r(n) = \sum_{q,s,t}\sum_{u,p} \mathbf{A}_{rq}(u)\mathbf{Q}^{(m)}_{qs}\mathbf{B}_{st}(p)\mathbf{x}_t(n-p-u),$$

where the range of the variables is $u \in \{0,\ldots,l_A\}$, $p \in \{0,\ldots,l_B\}$ and $r,q,s,t \in \{1,\ldots,R\}$. Let the output vector of the filter $\mathbf{B}(z)$ from Eq. (13) be denoted by $\mathbf{w}(n)$, then we also have the relation

$$\mathbf{y}_r(n) = \sum_{qs}\sum_{u} \mathbf{A}_{rq}(u)\mathbf{Q}^{(m)}_{qs}\mathbf{w}_s(n-u).$$

The fourth-order cumulants of $\mathbf{x}(n)$ will be denoted as

$$C_2^{2,\mathbf{x}}[r,\mathbf{k}] = \mathrm{Cum}[x_r(n-k_1), x_r^*(n-k_2), x_r(n-k_3), x_r^*(n-k_4)],$$

$$C_2^{2,\mathbf{x}}[\mathbf{r},\mathbf{k}] = \mathrm{Cum}[x_{r_1}(n-k_1), x_{r_2}^*(n-k_2), x_{r_3}(n-k_3), x_{r_4}^*(n-k_4)],$$

where $\mathbf{r} = [r_1,r_2,r_3,r_4]$ and $\mathbf{k} = [k_1,k_2,k_3,k_4]$. Under the assumption that the signs $\varepsilon = \pm 1$ of the kurtosis of the different signals are the same, it was shown in [7] that the function

$$\mathcal{J}_1^4 = \varepsilon \sum_{r=1}^{R} C_2^{2,\mathbf{y}}[r,\mathbf{0}] \tag{14}$$

is a contrast, where

$$C_2^{2,\mathbf{y}}[r,\mathbf{0}] = \sum_{\mathbf{s}}\sum_{\mathbf{u}}\sum_{\mathbf{q}} \mathbf{A}_{rq_1}(u_1)\mathbf{A}^*_{rq_2}(u_2)\mathbf{A}_{rq_3}(u_3)\mathbf{A}^*_{rq_4}(u_4)$$

$$\mathbf{Q}^{(m)}_{q_1s_1}\mathbf{Q}^{(m)*}_{q_2s_2}\mathbf{Q}^{(m)}_{q_3s_3}\mathbf{Q}^{(m)*}_{q_4s_4}C_2^{2,\mathbf{w}}[\mathbf{s},\mathbf{u}],$$

$$C_2^{2,\mathbf{w}}[\mathbf{s},\mathbf{u}] = \sum_{\mathbf{p}}\sum_{\mathbf{j}} \mathbf{B}_{s_1j_1}(p_1)\mathbf{B}^*_{s_2j_2}(p_2)\mathbf{B}_{s_3j_3}(p_3)\mathbf{B}^*_{s_4j_4}(p_4)C_2^{2,\mathbf{x}}[\mathbf{j},\mathbf{u}+\mathbf{p}]$$

and $\mathbf{s},\mathbf{q},\mathbf{j} \in \{1,\ldots,R\}^4$, $\mathbf{u} \in \{0,\ldots,l_A\}^4$ and $\mathbf{p} \in \{0,\ldots,l_B\}^4$.

For the two-channel case, $R=2$, the computation of the rotation matrix $\mathbf{Q}^{(m)}$ that will maximize (14) amounts to the rooting of a 56th degree polynomial [13] when a resultant-based procedure is employed. This polynomial generally does not admit more than 16 real roots, see [2,13] for further details. Hence, the PAFA algorithm must be considered as computationally more demanding than the PAJOD algorithm. For the case $R > 2$ no solution for the PAFA problem was given in [2].

3. Simulation results

The algorithms PAJOD, PAJOD2 and PAFA will be tested on random 2-Input-2-Output channels of varying SNR. The transmitted data blocks consist each of a QPSK sequence of 512 symbols length. The paraunitary channel is generated, just as in [1], as follows:

$$\mathbf{F}(z) = \mathbf{R}(\phi_0,\theta_0)\prod_{m=1}^{K-1}\mathbf{Z}(z)\mathbf{R}(\phi_m,\theta_m),$$

where

$$\mathbf{Z}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}, \quad \mathbf{R}(\phi_m,\theta_m) = \begin{bmatrix} \cos(\phi_m) & -\sin(\phi_m)\mathrm{e}^{-i\theta_m} \\ \sin(\phi_m)\mathrm{e}^{i\theta_m} & \cos(\phi_m) \end{bmatrix}$$

and the parameters ϕ_m and θ_m are drawn from a uniform distribution over $[0,2\pi)$. The filtered QPSK sequences of unit variance are perturbed by an additive white circular complex Gaussian noise with identity covariance matrix.

To see if the PAFA algorithm, which fully exploits the paraunitary structure of the channel, is better than the PAJOD algorithm, which only partially exploits the paraunitary structure of the channel, the performance will be based on the distance between the estimated and the true equalizer and the symbol error rate (SER). Furthermore, the computational complexity of the algorithms will also serve as a basis of comparison. To measure the elapsed time used to execute the algorithms in MATLAB, the built-in functions *tic*(·) and *toc*(·) are used.

Due to the inherent indeterminacies of the equalization problem, the equalizer can only be estimated up to a matrix of the form $\mathbf{D}(z) = \Lambda(z)\mathbf{P}$, where $\mathbf{P}$ is a permutation matrix and $\Lambda(z)$ is diagonal matrix with entries of the form λz^k, $k \in \mathbb{Z}$. Let $\mathbf{G}$ be the $R \times \mathrm{R}(2L-1)$ matrix obtained by stacking the global impulse response matrices after each other, then a measure of how far we are from perfect equalization is given by

$$\mathrm{dist}(\mathbf{G}) = \min_{\mathbf{D}(z)} \|\mathbf{G}\mathbf{D}(z)\|_F.$$

A procedure to solve this problem was given in [14] and will not be repeated here.

3.1. Paraunitary channel with $K=L$

In the first simulation, the paraunitary channel and equalizer are of length $K=L=3$. The mean SER, time and dist values over 100 simulations can be seen in Fig. 2. From Fig. 2(b) it is observed that below 15 dB the PAFA algorithm seems to capture the channel structure better than the PAJOD algorithm. It is also clear that the PAJOD2 algorithm is cheaper than the PAJOD algorithm which in turn is cheaper than the PAFA method. We also observe a variation of the computation time with the noise level. The reason is that at low SNR the problem of extracting the paraunitary channel from the cumulants of the observed data is more costly since the paraunitary structure is more "hidden" in the noisy data.

3.2. Paraunitary channel with $K<L$

A second simulation was conducted in order to investigate the robustness of the paraunitary method against channel order mismatch. We consider the case when $K=3$ and $L=4$, i.e., the channel is overmodeled. The mean SER, time and dist values over 100 simulations can be seen in Fig. 3. Comparison with Fig. 2 shows that the algorithms can handle the mismatch quite well. The relative performance of these algorithms remains the same.

3.3. Paraunitary channel with varying length K

A third simulation was conducted in order to investigate the computation time of the algorithms as a function of the filter length. The paraunitary channel and equalizer are of the same length $K=L$. The SNR was fixed to 10 while the filter and channel length varied from 4 to 10 with a hop factor of 1.

The mean SER and computation time over 100 simulations can be seen in Fig. 4. Here it is clear that the computational complexity of the PAFA algorithm grows exponentially as a function of the filter length. We also note that the gap in computation time between the PAJOD and PAJOD2 methods decreases with increasing K. The reason is that when K increases, then so does the time needed to compute the coefficients of the involved polynomials, while the actual cost of the rooting of the polynomials remains the same. However, the PAJOD method has to compute 24 coefficients while the PAJOD2 method only has to compute six coefficients. Consequently, a gap in computation time between the methods remains apparent. Secondly, the computation of the involved cumulant tensors also increases as illustrated in Fig. 5. In the case of slowly changing channels an adaptive estimation of the cumulant tensor is possible. Hence in that case a reduction of the computation time of the cumulants is possible.

4. Conclusion

In this paper the problem of blind equalization of paraunitary channels by means of Jacobi-type procedures was studied. After a review of the PAJOD and PAFA methods we proposed a computationally more efficient method called PAJOD2. The proposed method simplified

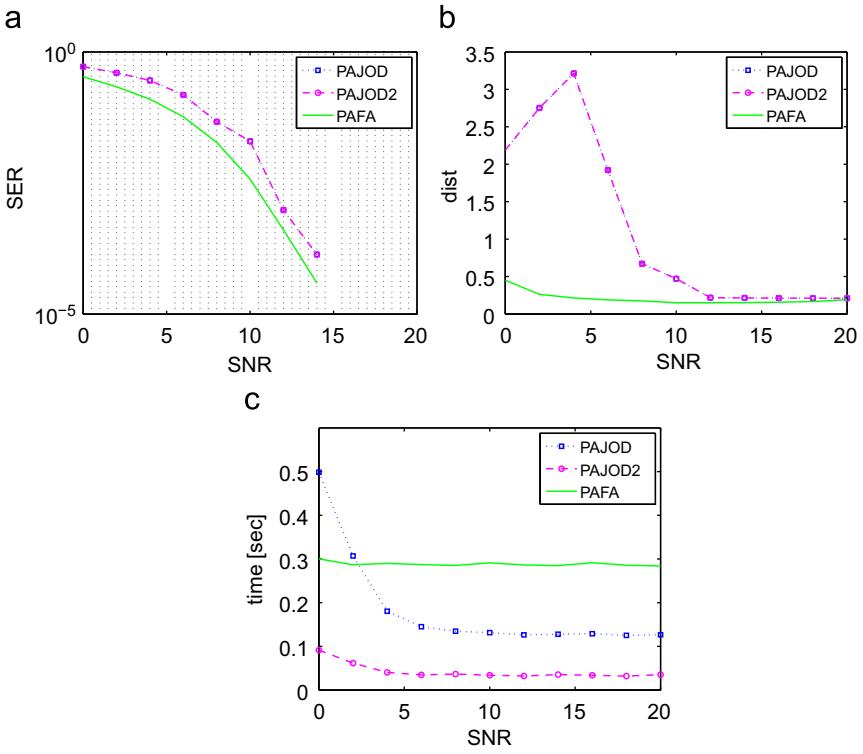


Fig. 2. Mean SER, dist and time values for the case when $K=L=3$. (a) Mean SER. (b) Mean dist. (c) Mean time.

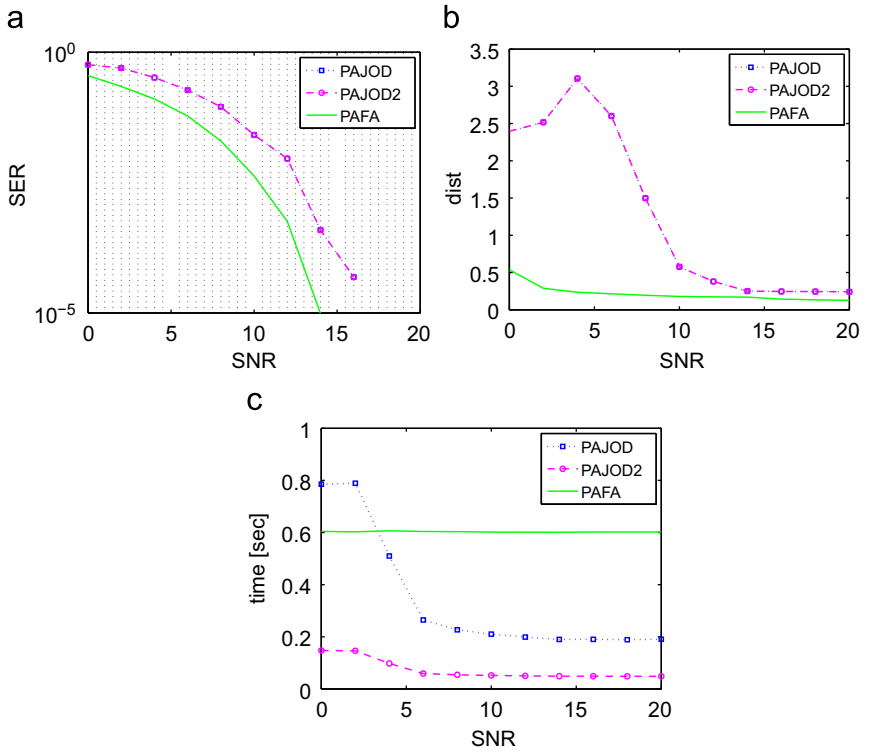


Fig. 3. Mean SER, dist and time values for the case when $K=3$ and $L=4$. (a) Mean SER. (b) Mean dist. (c) Mean time.

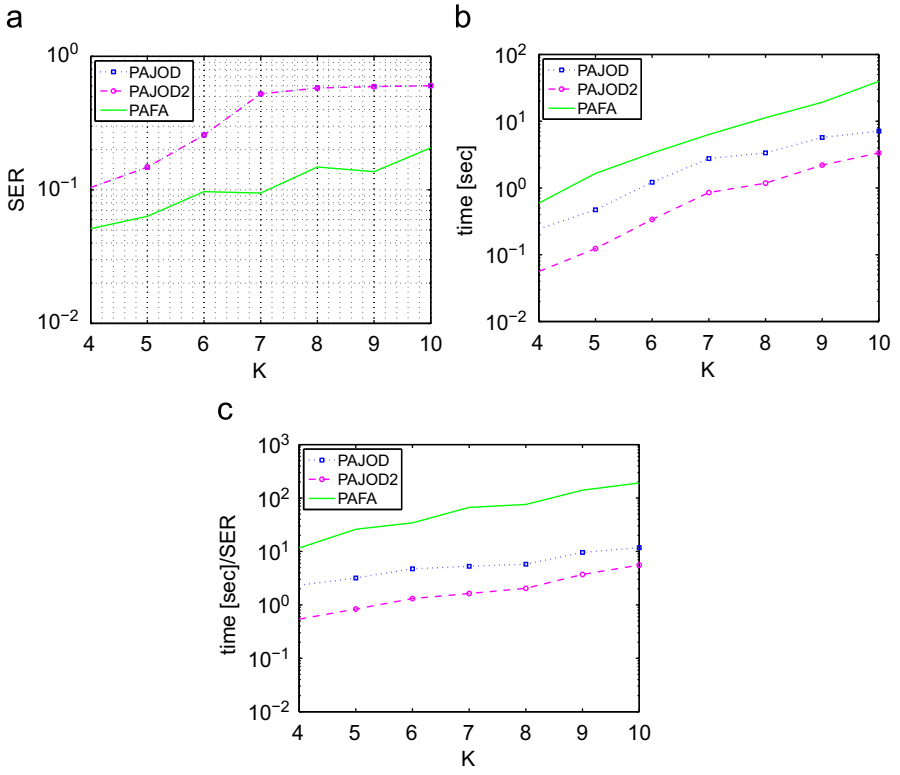


Fig. 4. Mean SER and computation time for filters and channels of varying length $K=L$. (a) Mean SER. (b) Mean dist. (c) Mean time.

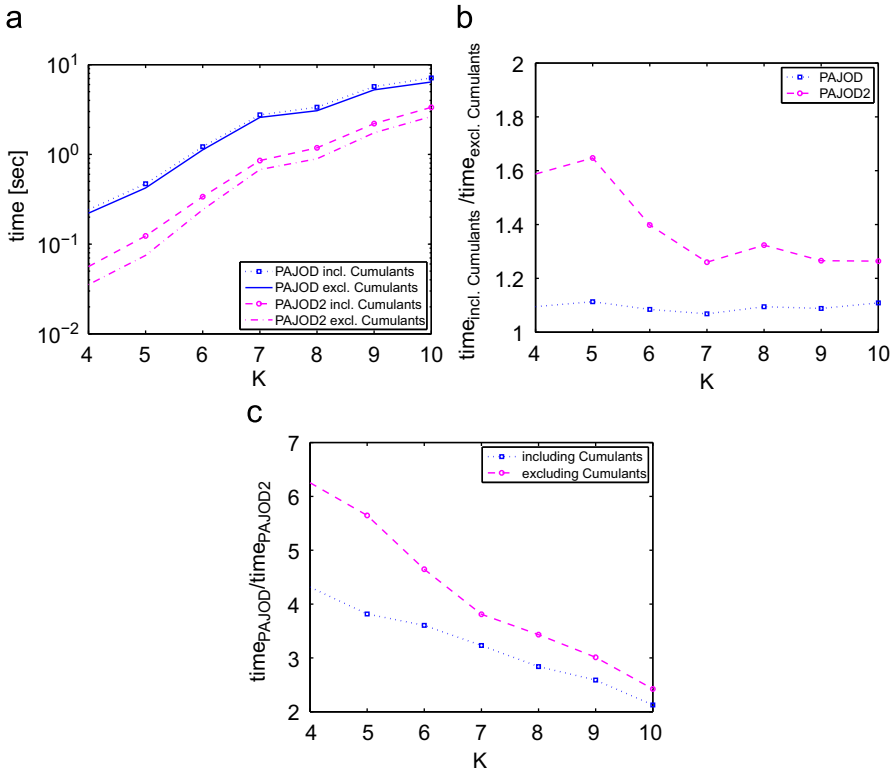


Fig. 5. Mean computation times for filters and channels of varying length $K=L$. (a) Mean time. (b) Mean $\text{time}_{\text{incl. Cumulants}}/\text{time}_{\text{excl. Cumulants}}$. (c) Mean $\text{time}_{\text{PAJOD}}/\text{time}_{\text{PAJOD2}}$.

the Jacobi-subproblem of the PAJOD method by going from the rooting of a 24th degree polynomial to the rooting of a polynomial of degree 6.

Simulation results indicate that for low SNR the PAFA method is more accurate than the PAJOD method. However, for high SNR both methods perform similarly. Such a comparison has not been conducted in the literature. Furthermore, computer simulations confirmed that the PAJOD2 method is faster than the PAJOD method.

Acknowledgments

Mikael Sørensen and Lieven De Lathauwer would like to thank L. Rota for kindly providing the PAJOD and PAFA Matlab code. Research supported by: (1) Research Council K.U.Leuven: GOA-MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1, STRT1/08/23, (2) F.W.O.: (a) project G.0427.10N, (b) Research Communities ICCoS, ANMMM and MLDM, (3) the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007–2011), (4) EU: ERNSI.

References

[1] P. Comon, L. Rota, Blind separation of independent sources from convolutive mixtures, IEICE Trans. Fund. (2003) 542–549.
[2] S. Icart, P. Comon, L. Rota, Blind paraunitary equalization, Signal Process. 89 (2009) 283–290.
[3] J.G. McWhirter, P.D. Baxter, T. Cooper, J. Foster, An EVD algorithm for para-Hermitian polynomial matrices, IEEE Trans. Signal Process. 5 (2007) 2158–2169.
[4] M. Sørensen, L. De Lathauwer, L. Deneire, An efficient Jacobi-type algorithm for blind equalization of paraunitary channels, in: Proceedings of the EUSIPCO 2009, pp. 510–514.
[5] P. Vaidyanathan, Multirate Systems and Filter Banks, Signal Processing Series, Prentice-Hall, New Jersey, 1993.
[6] D. Donoho, On minimum entropy deconvolution, Appl. Time Ser. Anal. II (1981) 565–608.
[7] P. Comon, Contrasts for multichannel blind deconvolution, IEEE Signal Process. Lett. 3 (1996) 209–211.
[8] J.-F. Cardoso, A. Souloumiac, Blind beamforming for non-Gaussian signals, IEE Proc. F 140 (1993) 362–370.
[9] J.-F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization, SIAM J. Math. Anal. Appl. 17 (1996) 161–164.
[10] G. Golub, C. Van Loan, Matrix Computations, 3rd ed. in 1996, The John Hopkins University Press, Baltimore, 1983.
[11] D. Cox, J. Little, D. O'Shea, Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 2nd ed. in 1996, Springer Verlag, 1992.
[12] A. Belouchrani, K. Abed-Meraim, Y. Hua, Jacobi-like algorithms for joint block diagonalization: applications to source localization, in: Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communications Systems (ISPACS), 1998.
[13] L. Rota, Egalisation aveugle de systemes multi-utilisateurs, Ph.D. Thesis, Laboratoire I3S, France, 2004.
[14] P. Comon, Blind Separation of Independent Sources from Convolutive Mixtures, A Partial Joint Diagonalization (PAJOD) Approach, Technical Report RR-2002-04-FR, I3S, 2002.