

Fast and stable YAST algorithm for principal and minor subspace tracking

Roland Badeau, *Member, IEEE*, Gaël Richard, *Senior Member, IEEE*, and Bertrand David, *Member, IEEE*

Abstract—This paper presents a new implementation of the YAST algorithm for principal and minor subspace tracking. YAST was initially derived from the Subspace Projection (SP) algorithm by C.E. Davila, which was known for its exceptional convergence rate, compared to other classical principal subspace trackers. The novelty in the YAST algorithm was the lower computational cost (linear if the data correlation matrix satisfies a so-called shift-invariance property), and the extension to minor subspace tracking. However, the original implementation of the YAST algorithm suffered from a numerical stability problem (the subspace weighting matrix slowly loses its orthonormality). We thus propose in this paper a new implementation of YAST, whose stability is established theoretically and tested via numerical simulations. This algorithm combines all the desired properties for a subspace tracker: remarkably high convergence rate, lowest steady state error, linear complexity, and numerical stability regarding the orthonormality of the subspace weighting matrix.

Index Terms—Adaptive estimation, Time series, Algorithms, Numerical stability, Convergence of numerical methods, Complexity theory.

I. INTRODUCTION

FAST estimation and tracking of the principal or minor subspace of a sequence of random vectors is a major problem in many applications, such as adaptive filtering, system identification, source localization, and spectral analysis for instance [1]. This problem can be stated as follows: considering a sequence of n -dimensional random data vectors $\{\mathbf{x}(t)\}_{t \in \mathbb{Z}}$, whose time-varying correlation matrix is denoted $\mathbf{C}_{xx}(t)$, we aim at estimating and tracking the r -dimensional principal or minor eigensubspace¹ of $\mathbf{C}_{xx}(t)$ (with $r < n$), which can be represented by its orthogonal projector $\mathbf{\Pi}(t)$, of rank r . The exact solution of this problem is well known: the principal (resp. minor) subspace of $\mathbf{C}_{xx}(t)$ is that which maximizes (resp. minimizes) the *generalized Rayleigh quotient*

$$\mathcal{J}(\mathbf{\Pi}(t)) = \text{trace}(\mathbf{C}_{xx}(t)\mathbf{\Pi}(t)). \quad (1)$$

Roland Badeau, Gaël Richard, and Bertrand David are with the Department of Signal and Image Processing, TELECOM ParisTech (ENST) - LTCI CNRS, Paris, France. E-mail: [roland.badeau, bertrand.david, gael.richard]@telecom-paristech.fr.

The research leading to this paper was supported by the ACI *Masse de données* Music Discover, and by the European Commission under contract FP6-027026, Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content - K-Space.

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

¹The principal (resp. minor) eigensubspace of $\mathbf{C}_{xx}(t)$ is defined as the subspace spanned by the r eigenvectors of $\mathbf{C}_{xx}(t)$ associated to r greatest (resp. lowest) eigenvalues.

The orthogonal projector $\mathbf{\Pi}(t)$ is generally parameterized by an $n \times r$ orthonormal matrix $\mathbf{W}(t)$, such that

$$\mathbf{\Pi}(t) = \mathbf{W}(t)\mathbf{W}(t)^H, \quad (2)$$

where the symbol H denotes the conjugate transpose of a matrix (or a vector). In practice, a straightforward calculation of this so-called *subspace weighting matrix* $\mathbf{W}(t)$ is computationally demanding: its complexity is $O(n^2r)$, which is inappropriate for real-time applications. Therefore the art of subspace tracking consists in recursively updating a matrix $\mathbf{W}(t)$, as close as possible to this exact solution, with a minimum of computations.

A large number of algorithms have been proposed for performing this task. Amidst this abundant literature², the PAST [3], LORAF3 [4], OPAST [5], and FAPI [6] algorithms belong to the family of Principal Subspace Trackers (PST), whereas the QRI [7] and HFRANS [8] algorithms belong to the family of Minor Subspace Trackers (MST). Besides, some algorithms such as ODKA [9], NOOJA [10], and FDPM [2], [11] can handle both principal and minor subspace tracking. The performance of these various algorithms is generally evaluated according to several criteria : their computational complexity, which is quantified as the number of operations performed at each iteration; the steady-state error, which is the mean estimation error when the algorithm has converged; and the convergence rate, which corresponds to the number of required iterations before reaching the steady state. Another desirable property for a subspace tracker is to guarantee the orthonormality of the subspace weighting matrix $\mathbf{W}(t)$ at each iteration. It is then worth evaluating the stability of the algorithm with respect to the orthonormality of $\mathbf{W}(t)$.

Recently, a new PST, referred to as the Subspace Projection (SP) algorithm was proposed by C.E. Davila [12]. We observed in [6] that the convergence rate of this algorithm is much higher than that of other classical PST's. However, this remarkable subspace tracker is computationally demanding. Indeed, the fastest implementation of SP relies on a so-called *shift invariance property* of the correlation matrix $\mathbf{C}_{xx}(t)$ (which typically holds in the case of time series analysis for instance), and has a complexity of $O(nr^2)$, whereas a number of existing subspace trackers only require $O(nr)$ operations per iteration (which is the case of all the above mentioned PST algorithms). Nevertheless, we found out that this drawback could be circumvented, and we proposed in [13] and [14] a new algorithm, referred to as YAST, which computes the same signal subspace as the SP algorithm, but only requires $O(nr)$ operations (when the correlation matrix $\mathbf{C}_{xx}(t)$ satisfies the

²A recent literature review can be found in [2] for instance.

shift invariance property). A version of YAST dedicated to MST was then proposed in [15].

Despite the great diversity of approaches proposed in the literature for performing PST and MST, most of the algorithms can be interpreted in the same unified framework: at each iteration t , the range space of $\mathbf{W}(t)$ belongs to a subspace of dimension $r+1$, spanned by $\mathbf{W}(t-1)$, plus one additional direction, which is often chosen as the data vector $\mathbf{x}(t)$ (this is the case of all the above mentioned algorithms for instance, although some of them involve several additional directions at each iteration). From this perspective, the exceptional convergence rate of SP and YAST can be easily explained: at each iteration, those two algorithms compute the "best" r -dimensional subspace inside this augmented subspace, *i.e.* the subspace which explicitly maximizes (or minimizes) the generalized Rayleigh quotient. Several versions of YAST have already been presented in the literature:

- Various additional search directions have been proposed, such as $\mathbf{x}(t)$ and $\mathbf{C}_{xx}(t-1)\mathbf{x}(t)$ in [13], [15]. A conjugate gradient method was also presented in [16], for computing an appropriate search direction in the case of MST. This method can easily be adapted to PST.
- Several window shapes were proposed for estimating the correlation matrix $\mathbf{C}_{xx}(t)$: an exponential window in [13], [15], and a truncated (or sliding) window in [14].

Unfortunately, it was observed in [17] and [18] that the implementations of the YAST PST presented in [13] and [15] suffer from a numerical stability problem (the subspace weighting matrix slowly loses its orthonormality). In order to solve this problem, we propose in this paper to apply a new orthogonalization procedure to the YAST algorithm, which is similar to that proposed in [2] for the FDPMP subspace tracker. The stability of the resulting algorithm is established theoretically and tested via numerical simulations. To keep the mathematical developments as concise as possible, the new implementation of YAST introduced in this paper thus involves the following changes with respect to earlier versions:

- this implementation stands for both principal and minor subspace tracking;
- there is only one additional search direction, which is the data vector $\mathbf{x}(t)$;
- the update of the correlation matrix is based on the simplest window shape (exponential window);
- this implementation involves an orthogonalization procedure similar to that proposed in [2].

Nevertheless, this implementation can be easily generalized to truncated windows and other search directions (*cf.* [13]–[16]). We also present a convergence analysis of the YAST algorithm, and we prove the numerical stability of the new implementation. The paper is organized as follows: in section II, the basic principle of the YAST algorithm is summarized. Then a fast implementation of YAST is proposed in section III. The performance of this subspace tracker is illustrated in section IV. The main conclusions of this paper are summarized in section V. Finally, the convergence and the numerical stability of the YAST algorithm are investigated in the Appendix.

II. OVERVIEW OF THE YAST ALGORITHM

A. Principle

Consider a sequence of independent n -dimensional random data vectors $\{\mathbf{x}(t)\}_{t \in \mathbb{Z}}$, whose $n \times n$ correlation matrix $\mathbf{C}_{xx}(t)$ is estimated by applying an exponential window to the data, leading to the update

$$\mathbf{C}_{xx}(t) = \beta \mathbf{C}_{xx}(t-1) + \mathbf{x}(t) \mathbf{x}(t)^H \quad (3)$$

where $0 < \beta < 1$ is the exponential forgetting factor. As mentioned in the introduction, the generalized Rayleigh quotient is maximized (resp. minimized) when the subspace weighting matrix $\mathbf{W}(t)$ spans the principal subspace (resp. the minor subspace) of $\mathbf{C}_{xx}(t)$. Unfortunately, implementing this optimization over all orthonormal matrices is computationally demanding, and does not lead to a simple recursion between $\mathbf{W}(t)$ and $\mathbf{W}(t-1)$. In order to reduce the complexity, the implementation of YAST proposed below recursively performs this search within the $(r+1)$ -dimensional subspace spanned by $\mathbf{W}(t-1)$ and $\mathbf{x}(t)$.

Let $\underline{\mathbf{\Pi}}(t)$ be the orthogonal projector on the augmented subspace, and $\underline{\mathbf{W}}(t)$ an $n \times (r+1)$ orthonormal matrix such that

$$\underline{\mathbf{\Pi}}(t) = \underline{\mathbf{W}}(t) \underline{\mathbf{W}}(t)^H. \quad (4)$$

Then any r -dimensional subspace of $\text{span}(\underline{\mathbf{\Pi}}(t))$ can be represented by the orthogonal projector

$$\mathbf{\Pi}(t) = \underline{\mathbf{\Pi}}(t) - \underline{\mathbf{v}}(t) \underline{\mathbf{v}}(t)^H, \quad (5)$$

where the unitary vector $\underline{\mathbf{v}}(t)$ belongs to the range space of $\underline{\mathbf{\Pi}}(t)$. In particular, $\underline{\mathbf{v}}(t)$ can be written in the form

$$\underline{\mathbf{v}}(t) = \underline{\mathbf{W}}(t) \underline{\phi}(t), \quad (6)$$

where $\underline{\phi}(t)$ is an $(r+1)$ -dimensional unitary vector. Substituting equations (4) to (6) into equation (1), the criterion to be optimized can be rewritten as

$$\mathcal{J}(\mathbf{\Pi}(t)) = \text{trace}(\underline{\mathbf{C}}_{yy}(t)) - \underline{\phi}(t)^H \underline{\mathbf{C}}_{yy}(t) \underline{\phi}(t) \quad (7)$$

where $\underline{\mathbf{C}}_{yy}(t)$ is the $(r+1) \times (r+1)$ matrix

$$\underline{\mathbf{C}}_{yy}(t) = \underline{\mathbf{W}}(t)^H \mathbf{C}_{xx}(t) \underline{\mathbf{W}}(t). \quad (8)$$

In this manner, the optimization of \mathcal{J} with respects to the $n \times r$ orthonormal matrix $\mathbf{W}(t)$ is replaced by the optimization of \mathcal{J} with respects to the $(r+1)$ -dimensional unitary vector $\underline{\phi}(t)$. The result of this optimization is well-known: according to equation (7), \mathcal{J} is maximized (resp. minimized) when $\underline{\phi}(t)$ is the minor (resp. principal) eigenvector of $\underline{\mathbf{C}}_{yy}(t)$. Finally, given the new data vector $\mathbf{x}(t)$, the YAST algorithm updates the previous subspace weighting matrix $\mathbf{W}(t-1)$ by successively computing

- 1) an orthonormal basis $\underline{\mathbf{W}}(t)$ of the augmented subspace;
- 2) the compressed matrix $\underline{\mathbf{C}}_{yy}(t)$ defined in equation (8);
- 3) the principal or minor eigenvector $\underline{\phi}(t)$ of $\underline{\mathbf{C}}_{yy}(t)$ (for MST or PST resp.), and a basis $\mathbf{W}(t)$ of the range space of the projector $\mathbf{\Pi}(t)$ defined in equation (5);
- 4) the $r \times r$ compressed matrix $\mathbf{C}_{yy}(t)$, defined as

$$\mathbf{C}_{yy}(t) = \mathbf{W}(t)^H \mathbf{C}_{xx}(t) \mathbf{W}(t), \quad (9)$$

which will permit a fast calculation of $\underline{\mathbf{C}}_{yy}(t)$ in step 2).

The convergence of this YAST algorithm is proved in Appendix B page 8, and a fast implementation is proposed in section III.

B. Numerical stability

As mentioned in the introduction, it was shown in [17] that the implementation of the YAST PST presented in [13] suffers from a numerical stability problem. Indeed, although YAST was originally designed to compute an orthonormal subspace weighting matrix $\mathbf{W}(t)$ at each iteration, when the estimated subspace gets too close to the exact subspace (which happens in the case of stationary data with a high signal-to-noise ratio), a sudden loss of orthogonality makes the algorithm diverge. The solution proposed in [17] to cope with this problem consisted in switching to another subspace tracker such as FAPI [6] when this critical point was reached. In the context of MST [15], we addressed the problem in a different way, by modifying our implementation of YAST in order to restore the orthogonality of the subspace weighting matrix at each iteration. Although this modification prevents the sudden divergence of YAST, it was proved in [18] that the accumulation of rounding errors still leads to a loss of orthogonality of the subspace weighting matrix, at a much lower rate however. In order to alleviate this effect, it was proposed in [18] to use a partial orthogonalization scheme called Pairwise Gram-Schmidt (PGS), which improves the numerical stability of YAST without increasing its computational cost. However, we will show in section IV an example of instable behavior of YAST-PGS.

In fact, the loss of orthogonality is a shortcoming common to most of the existing MST's of linear complexity. In particular, the classical Oja algorithm [19] is known to diverge. A very classical approach for improving the numerical stability of MST consists in using Householder transforms [8]–[10]. However this technique does not completely prevent the slow accumulation of rounding errors, as observed in [2], and the convergence rate of these algorithms is much lower than that of the classical PST techniques. To the best of our knowledge, the first MST of linear complexity which guaranteed a perfect stability with respect to the orthogonality was the Fast Data Projection Method (FDPM) by G. Doukopoulos and V. Moustakides [2], [11]. This algorithm relies on a specific orthogonalization procedure, whose numerical stability was theoretically proved. Considering the exceptional stability of FDPM, we propose in the following section to apply a similar orthogonalization procedure to the YAST algorithm, in order to solve the problems observed in our previous implementations. The numerical stability of this new implementation is proved in Appendix C page 9.

III. FAST IMPLEMENTATION OF YAST

Below, a fast implementation of the YAST algorithm is proposed, whose global cost is only $5nr$ flops³. Following the structure proposed in section II-A, it is composed of four steps: computation of $\underline{\mathbf{W}}(t)$ (section III-A), computation of $\underline{\mathbf{C}}_{yy}(t)$

(section III-B), update of $\mathbf{W}(t)$ (section III-C), and update of $\mathbf{C}_{yy}(t)$ (section III-D). The computational complexity of this implementation is addressed in section III-E.

A. Computation of $\underline{\mathbf{W}}(t)$

Define the r -dimensional compressed data vector

$$\mathbf{y}(t) = \mathbf{W}(t-1)^H \mathbf{x}(t). \quad (10)$$

Then the n -dimensional vector

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{W}(t-1) \mathbf{y}(t) \quad (11)$$

is orthogonal to the range space of $\mathbf{W}(t-1)$. Let $\sigma(t)$ be its Hermitian norm:

$$\sigma(t) = \|\mathbf{e}(t)\|_2. \quad (12)$$

If $\sigma(t) = 0$, then the new search direction $\mathbf{x}(t)$ is actually included in the old subspace spanned by $\mathbf{W}(t-1)$. This means that the new subspace is equal to the old one, thus we can simply choose $\mathbf{W}(t) = \mathbf{W}(t-1)$, and the iteration is over. Otherwise $\sigma(t) \neq 0$, and we can define the normalized vector

$$\mathbf{u}(t) = \mathbf{e}(t) / \sigma(t). \quad (13)$$

Then the $n \times (r+1)$ augmented matrix

$$\underline{\mathbf{W}}(t) = \left[\mathbf{W}(t-1) \mid \mathbf{u}(t) \right] \quad (14)$$

defines an orthonormal basis of $\text{span}([\mathbf{W}(t-1), \mathbf{x}(t)])$.

B. Computation of $\underline{\mathbf{C}}_{yy}(t)$

Substituting equation (3) and equations (9) to (14) into equation (8) yields

$$\underline{\mathbf{C}}_{yy}(t) = \left[\begin{array}{c|c} \mathbf{C}'_{yy}(t) & \mathbf{z}(t) \\ \hline \mathbf{z}(t)^H & \gamma(t) \end{array} \right], \quad (15)$$

where

$$\mathbf{C}'_{yy}(t) = \beta \mathbf{C}_{yy}(t-1) + \mathbf{y}(t) \mathbf{y}(t)^H \quad (16)$$

$$\mathbf{z}(t) = \beta \mathbf{W}(t-1)^H \mathbf{C}_{xx}(t-1) \mathbf{u}(t) + \sigma(t) \mathbf{y}(t) \quad (17)$$

$$\gamma(t) = \beta \mathbf{u}(t)^H \mathbf{C}_{xx}(t-1) \mathbf{u}(t) + \sigma(t)^2. \quad (18)$$

In order to reduce the computational cost, it will be useful to remark that $\mathbf{z}(t)$ and $\gamma(t)$ can be efficiently computed as

$$\mathbf{z}(t) = \beta \frac{\mathbf{y}''(t) - \mathbf{y}'(t)}{\sigma(t)} + \sigma(t) \mathbf{y}(t), \quad (19)$$

$$\gamma(t) = \beta \frac{\mathbf{x}(t)^H \mathbf{x}'(t) - 2\Re(\mathbf{y}(t)^H \mathbf{y}''(t)) + \mathbf{y}(t)^H \mathbf{y}'(t)}{\sigma(t)^2} + \sigma(t)^2 \quad (20)$$

where we have introduced the auxiliary vectors

$$\mathbf{x}'(t) = \mathbf{C}_{xx}(t-1) \mathbf{x}(t), \quad (21)$$

$$\mathbf{y}'(t) = \mathbf{C}_{yy}(t-1) \mathbf{y}(t), \quad (22)$$

$$\mathbf{y}''(t) = \mathbf{W}(t-1)^H \mathbf{x}'(t). \quad (23)$$

³In this paper, a flop is a multiply / accumulate (MAC) operation.

C. Update of $\mathbf{W}(t)$

Let $\underline{\phi}(t)$ be the unitary eigenvector associated to the greatest (or lowest) eigenvalue of the matrix $\underline{\mathbf{C}}_{yy}(t)$ of dimension $(r+1) \times (r+1)$. The YAST algorithm consists in estimating the new subspace as the orthogonal complement of the vector $\underline{\mathbf{v}}(t) = \underline{\mathbf{W}}(t)\underline{\phi}(t)$ (defined in equation (6)) in the range space of $\underline{\mathbf{W}}(t)$. To this end, we will first introduce a non-orthonormal matrix $\mathbf{T}(t)$ which spans this new subspace, then we will describe how this matrix can be orthonormalized.

1) *Computation of the non-orthonormal subspace weighting matrix $\mathbf{T}(t)$* : Vector $\underline{\phi}(t)$ is now decomposed in the form

$$\underline{\phi}(t) = \left[\frac{\varepsilon(t)\phi(t)}{\varphi(t)} \right] \theta(t), \quad (24)$$

where $\varphi(t)\theta(t)$ is the polar form of the last coefficient of $\underline{\phi}(t)$ (i.e. $\varphi(t) \in \mathbb{R}$ is non-negative, and $\theta(t)$ is a complex number such that $|\theta(t)| = 1$), $\varepsilon(t) \geq 0$, and $\phi(t)$ is an r -dimensional unitary vector. In particular, $\varepsilon(t)$ and $\varphi(t)$ are such that $\varepsilon(t)^2 + \varphi(t)^2 = 1$. The following lemma which involves $\varepsilon(t)$ is proved in Appendix A page 8.

Lemma III-C.1: $\varepsilon(t)$ is bounded by the distance between the estimated subspaces at times t and $t-1$, in the sense that

$$\varepsilon(t)^2 \leq \|\mathbf{\Pi}(t) - \mathbf{\Pi}(t-1)\|_2$$

where $\|\cdot\|_2$ denotes the matrix spectral norm, or 2-norm.

Lemma III-C.1 shows that when the algorithm has reached its steady state, i.e. when the subspace admits slow variations ($\|\mathbf{\Pi}(t) - \mathbf{\Pi}(t-1)\|_2 \ll 1$), then $\varepsilon(t)$ is much smaller than 1.

Next, the following lemma introduces the non-orthonormal subspace weighting matrix $\mathbf{T}(t)$.

Lemma III-C.2: The range space of the $n \times r$ matrix

$$\mathbf{T}(t) = \mathbf{W}(t-1) - \varepsilon(t)\mathbf{u}(t)\phi(t)^H \quad (25)$$

is a subspace of $\text{span}(\underline{\mathbf{W}}(t))$, such that

$$\|\underline{\mathbf{v}}(t)^H \mathbf{T}(t)\|_2 \leq (\varepsilon(t))^3. \quad (26)$$

Lemma III-C.2 shows that when $\varepsilon(t) \rightarrow 0$, the range space of $\mathbf{T}(t)$ asymptotically matches the orthogonal complement of $\underline{\mathbf{v}}(t)$ in the augmented subspace $\text{span}(\underline{\mathbf{W}}(t))$. Lemma III-C.2 is also proved in Appendix A page 8.

2) *Computation of the orthonormal subspace weighting matrix $\mathbf{W}(t)$* : The following developments aim at orthonormalizing $\mathbf{T}(t)$ to obtain the new subspace weighting matrix $\mathbf{W}(t)$ ⁴. In order to ensure the numerical stability of our algorithm, the proposed orthonormalization method is similar to that presented in [2]. First, equation (25) yields

$$\mathbf{T}(t)^H \mathbf{T}(t) = \mathbf{I}_r + \varepsilon(t)^2 \phi(t) \phi(t)^H. \quad (27)$$

(here \mathbf{I}_r denotes the $r \times r$ identity matrix). Define the r -dimensional unitary vector

$$\mathbf{e}_1(t) = -e^{i \text{angle}(\phi_1(t))} [1, 0 \dots 0]^T, \quad (28)$$

⁴Actually, an exact implementation of YAST would require that $\mathbf{T}(t)$ be exactly orthogonal to $\underline{\mathbf{v}}(t)$. However we observed in practice that this (negligible) approximation improves both the stability and the convergence rate of YAST, as illustrated in section IV-B2.

where the symbol T denotes the transpose of a vector (or a matrix), and $\phi_1(t)$ is the first coefficient of $\phi(t)$. Then define the r -dimensional unitary vector

$$\mathbf{a}(t) = \frac{\phi(t) - \mathbf{e}_1(t)}{\|\phi(t) - \mathbf{e}_1(t)\|_2}, \quad (29)$$

and consider the Householder matrix

$$\mathbf{S}(t) = \mathbf{I}_r - 2\mathbf{a}(t)\mathbf{a}(t)^H, \quad (30)$$

which transforms the vector $\phi(t)$ into $\mathbf{e}_1(t)$. Note that the denominator in the right member of equation (29) can never be zero (more precisely, it belongs to the interval $[1, 2]$, which prevents a division by zero). Let

$$\mathbf{Q}(t) = \mathbf{T}(t) \mathbf{S}(t). \quad (31)$$

Then equations (27) and (31) yield

$$\mathbf{Q}(t)^H \mathbf{Q}(t) = \mathbf{I}_r + \varepsilon(t)^2 \mathbf{e}_1(t) \mathbf{e}_1(t)^H,$$

which shows that the columns of the matrix $\mathbf{Q}(t)$ are orthogonal. Therefore an orthonormal matrix $\mathbf{W}(t)$ can be obtained by normalizing the first column of $\mathbf{Q}(t)$:

$$\mathbf{W}(t) = \mathbf{Q}(t) \mathbf{D}(t) \quad (32)$$

where $\mathbf{D}(t)$ is the $r \times r$ diagonal matrix

$$\mathbf{D}(t) = \text{diag}(1/\|\mathbf{q}_1(t)\|_2, 1 \dots 1), \quad (33)$$

$\mathbf{q}_1(t)$ being the first column of $\mathbf{Q}(t)$. Finally, substituting equations (25) and (30) into equation (31) yields

$$\mathbf{Q}(t) = \mathbf{W}(t-1) - 2\mathbf{b}(t)\mathbf{a}(t)^H - \varepsilon(t)\mathbf{u}(t)\mathbf{e}_1(t)^H \quad (34)$$

where

$$\mathbf{b}(t) = \mathbf{W}(t-1) \mathbf{a}(t). \quad (35)$$

D. Update of $\mathbf{C}_{yy}(t)$

The auxiliary matrix $\mathbf{C}_{yy}(t) = \mathbf{W}(t)^H \mathbf{C}_{xx}(t) \mathbf{W}(t)$ defined in equation (9) can also be efficiently updated. Indeed, substituting equations (31), (30), (25), and (14) into equation (32) yields

$$\mathbf{W}(t) = \underline{\mathbf{W}}(t) \mathbf{U}(t) \mathbf{D}(t) \quad (36)$$

where the $(r+1) \times r$ matrix $\mathbf{U}(t)$ is defined as

$$\mathbf{U}(t) = \begin{bmatrix} \mathbf{I}_r - 2\mathbf{a}(t)\mathbf{a}(t)^H \\ -\varepsilon(t)\mathbf{e}_1(t)^H \end{bmatrix}. \quad (37)$$

Then substituting equations (8) and (36) into equation (9) yields

$$\mathbf{C}_{yy}(t) = \mathbf{D}(t) \mathbf{C}_{yy}''(t) \mathbf{D}(t). \quad (38)$$

where

$$\mathbf{C}_{yy}''(t) = \mathbf{U}(t)^H \underline{\mathbf{C}}_{yy}(t) \mathbf{U}(t). \quad (39)$$

Finally, substituting equations (15) and (37) into equation (39) yields

$$\mathbf{C}_{yy}''(t) = \text{Herm}(\mathbf{C}'_{yy}(t) - \mathbf{a}'(t)\mathbf{a}(t)^H - \varepsilon(t)\mathbf{z}'(t)\mathbf{e}_1(t)^H) \quad (40)$$

where $\text{Herm}(\cdot)$ denotes the Hermitian part of a matrix ($\text{Herm}(\mathbf{M}) = \frac{\mathbf{M} + \mathbf{M}^H}{2}$), and

$$\mathbf{a}'(t) = 4\mathbf{C}'_{yy}(t)\mathbf{a}(t) - 4(\mathbf{a}(t)^H \mathbf{C}'_{yy}(t)\mathbf{a}(t))\mathbf{a}(t), \quad (41)$$

$$\mathbf{z}'(t) = 2\mathbf{z}(t) - 4(\mathbf{a}(t)^H \mathbf{z}(t))\mathbf{a}(t) - \varepsilon(t)\gamma(t)\mathbf{e}_1(t). \quad (42)$$

TABLE I
PSEUDO-CODE OF THE YAST ALGORITHM

	eq.:	flops:
Input: $\mathbf{x}(t)$		
$\mathbf{y}(t) = \mathbf{W}(t-1)^H \mathbf{x}(t)$	(10)	nr
$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{W}(t-1)\mathbf{y}(t)$	(11)	nr
$\sigma(t) = \ \mathbf{e}(t)\ _2$	(12)	n
if $\sigma(t) \neq 0$,		
$\mathbf{u}(t) = \mathbf{e}(t) / \sigma(t)$	(13)	
$\mathbf{x}'(t) = \mathbf{C}_{xx}(t-1)\mathbf{x}(t)$	(21)	$9n$
$\mathbf{y}'(t) = \mathbf{C}_{yy}(t-1)\mathbf{y}(t)$	(22)	r^2
$\mathbf{y}''(t) = \mathbf{W}(t-1)^H \mathbf{x}'(t)$	(23)	nr
$\mathbf{C}'_{yy}(t) = \beta \mathbf{C}_{yy}(t-1) + \mathbf{y}(t)\mathbf{y}(t)^H$	(16)	r^2
$\mathbf{z}(t) = \beta (\mathbf{y}''(t) - \mathbf{y}'(t)) / \sigma(t) + \sigma(t)\mathbf{y}(t)$	(19)	r
$\gamma(t) = \beta \frac{\mathbf{x}(t)^H \mathbf{x}'(t) - 2\Re(\mathbf{y}(t)^H \mathbf{y}''(t)) + \mathbf{y}(t)^H \mathbf{y}'(t)}{\sigma(t)^2} + \sigma(t)^2$	(20)	n
$\underline{\mathbf{C}}_{yy}(t) = \begin{bmatrix} \mathbf{C}'_{yy}(t) & \mathbf{z}(t) \\ \mathbf{z}(t)^H & \gamma(t) \end{bmatrix}$	(15)	
if Principal Subspace Tracking		
$(\phi(t), \lambda(t)) = \min \{ \text{eig}(\underline{\mathbf{C}}_{yy}(t)) \}$		$O(r^2)$
else if Minor Subspace Tracking		
$(\phi(t), \lambda(t)) = \max \{ \text{eig}(\underline{\mathbf{C}}_{yy}(t)) \}$		$O(r^2)$
end if		
$[\varepsilon(t)\phi(t)^T, \varphi(t)] \theta(t) = \phi(t)^T$	(24)	$2r$
$\mathbf{e}_1(t) = -e^{i \angle \phi(t)} [1, 0 \dots 0]^T$	(28)	
$\mathbf{a}(t) = \frac{\phi(t) - \mathbf{e}_1(t)}{\ \phi(t) - \mathbf{e}_1(t)\ _2}$	(29)	r
$\mathbf{b}(t) = \mathbf{W}(t-1)\mathbf{a}(t)$	(35)	nr
$\mathbf{Q}(t) = \mathbf{W}(t-1) - 2\mathbf{b}(t)\mathbf{a}(t)^H - \varepsilon(t)\mathbf{u}(t)\mathbf{e}_1(t)^H$	(34)	nr
$\mathbf{D}(t) = \text{diag}(1/\ \mathbf{q}_1(t)\ _2, 1 \dots 1)$	(33)	n
$\mathbf{W}(t) = \mathbf{Q}(t)\mathbf{D}(t)$	(32)	
$\mathbf{a}'(t) = 4\mathbf{C}'_{yy}(t)\mathbf{a}(t) - 4(\mathbf{a}(t)^H \mathbf{C}'_{yy}(t)\mathbf{a}(t))\mathbf{a}(t)$	(41)	r^2
$\mathbf{z}'(t) = 2\mathbf{z}(t) - 4(\mathbf{a}(t)^H \mathbf{z}(t))\mathbf{a}(t) - \varepsilon(t)\gamma(t)\mathbf{e}_1(t)$	(42)	$2r$
$\mathbf{C}''_{yy}(t) = \text{Herm}(\mathbf{C}'_{yy}(t) - \mathbf{a}'(t)\mathbf{a}(t)^H - \varepsilon(t)\mathbf{z}'(t)\mathbf{e}_1(t)^H)$	(39)	$2r^2$
$\mathbf{C}_{yy}(t) = \mathbf{D}(t)\mathbf{C}''_{yy}(t)\mathbf{D}(t)$	(40)	
end if		
Total:		$5nr + O(r^2 + n)$

E. Computational complexity

The complete pseudo-code of YAST is summarized in table I⁵. Note that this implementation requires computing the principal or minor⁶ eigenvector $\phi(t)$ of the compressed matrix $\underline{\mathbf{C}}_{yy}(t)$. This operation can be performed by means of a recursive algorithm, such as the *shifted* power iteration method [20], whose dominant cost is r^2 flops per iteration. This leads to an overall complexity of $O(r^2)$ if we assume that the number of performed iterations is much lower than r , or $O(r^3)$ if we consider that this number is proportional to r . In order to reduce the number of performed iterations, since $\varepsilon(t)$ is small when the algorithm has reached its steady state (as mentioned in section III-C), we propose to choose the vector $[0 \dots 0, 1]^T$ as an initial guess for starting this recursion. Alternately, we introduced in [16] a conjugate gradient (CG) algorithm for computing the minor eigenvector of a positive semidefinite matrix, which presents a much faster convergence rate than the power iteration method, for the same computational cost. The pseudo-code of this CG algorithm applied to the computation of the minor or principal eigenvector of $\underline{\mathbf{C}}_{yy}(t)$ is presented in table II, with a proper initialization. In practice, we observed

⁵In table I, the computational costs of multiplications and divisions of complex numbers by positive numbers have been neglected.

⁶In the case of PST, this pseudo-code involves the computation of the minor eigenvector of $\underline{\mathbf{C}}_{yy}(t)$. Since this operation is generally more complex than computing the principal eigenvector, an other possible implementation of the YAST PST algorithm consists in recursively updating the inverse of $\underline{\mathbf{C}}_{yy}(t)$, and computing $\phi(t)$ as the principal eigenvector of this matrix, as initially proposed in [13].

that the performance of the YAST algorithm was unaltered if one iteration only of the CG algorithm is performed⁷, which permits to compute $\phi(t)$ with an overall complexity of $O(r^2)$ flops only.

In other respects, the calculation cost of the vector $\mathbf{x}'(t) = \mathbf{C}_{xx}(t-1)\mathbf{x}(t)$ is normally n^2 flops. However, if the data correlation matrix satisfies a shift invariance property, this cost can be reduced to $9n$ flops by means of the technique described in [12]. Therefore the global cost of the YAST algorithm is reduced to $5nr$ flops. This cost can then be compared to that of other existing subspace trackers. In the PST category for instance, the complexity of PAST [3] and FAPI [6] is $3nr$, that of OPAST [5] is $4nr$, that of LORAF3 [21] is $18nr$, and that of SP1 [12] is $4nr^2$. In the MST category, the complexity of HFRANS [8] and NOOJA [10] is $4nr$, that of FDPM [2], [11] and YAST-PGS [18] is $5nr$, that of ODKA [9] is $7nr$, and that of QRI [7] is n^2r .

TABLE II
PSEUDO-CODE OF THE CONJUGATE GRADIENT ALGORITHM

	flops:
Initialization of the CG algorithm at time t	
$\mathbf{g} = \mathbf{z}(t) / \ \mathbf{z}(t)\ $	r
$\mathbf{p} = \mathbf{C}'_{yy}(t)\mathbf{g} - (\mathbf{g}^H \mathbf{C}'_{yy}(t)\mathbf{g})\mathbf{g}$	r^2
$\mathbf{S}(0) = \begin{bmatrix} \mathbf{0}_r & \mathbf{p} / \ \mathbf{p}\ & \mathbf{g} \\ 1 & 0 & 0 \end{bmatrix}$	r
$\underline{\mathbf{C}}_{ys}(0) = \underline{\mathbf{C}}_{yy}(t)\mathbf{S}(0)$	r^2
$\mathbf{C}_{ss}(0) = \mathbf{S}(0)^H \underline{\mathbf{C}}_{ys}(0)$	$6r$
Repeat at each iteration $k \geq 1$	
if minor eigenvector computation	
$(\theta(k), \lambda(k)) = \min \{ \text{eig}(\mathbf{C}_{ss}(k-1)) \}$	$O(1)$
else if principal eigenvector computation	
$(\theta(k), \lambda(k)) = \max \{ \text{eig}(\mathbf{C}_{ss}(k-1)) \}$	$O(1)$
end if	
$\vartheta(k) = \begin{bmatrix} -\sqrt{ \theta_2(k) ^2 + \theta_3(k) ^2} \\ \theta_1^*(k) \frac{\theta_2(k)}{ \theta_2(k) } / \sqrt{1 + \left \frac{\theta_3(k)}{\theta_2(k)} \right ^2} \\ \theta_1^*(k) \frac{\theta_3(k)}{ \theta_3(k) } / \sqrt{1 + \left \frac{\theta_2(k)}{\theta_3(k)} \right ^2} \end{bmatrix}$	$O(1)$
$\Theta(k) = [\theta(k), \vartheta(k)]$	
$\mathbf{S}(k)_{(:,1:2)} = \mathbf{S}(k-1)\Theta(k)$	$6r$
$\underline{\mathbf{C}}_{ys}(k)_{(:,1:2)} = \underline{\mathbf{C}}_{ys}(k-1)\Theta(k)$	$6r$
$\mathbf{C}_{ss}(k)_{(1:2,1:2)} = \Theta(k)^H \mathbf{C}_{ss}(k-1)\Theta(k)$	12
$\frac{1}{2} \nabla J(k) = \underline{\mathbf{C}}_{ys}(k)_{(:,1)} - \lambda(k)\mathbf{S}(k)_{(:,1)}$	r
$\mathbf{g}(k) = \nabla J(k) / \ \nabla J(k)\ $	r
$\mathbf{g}(k) \leftarrow \mathbf{g}(k) - \mathbf{S}(k)_{(:,1:2)} \left(\mathbf{S}(k)_{(:,1:2)}^H \mathbf{g}(k) \right)$	$4r$
$\mathbf{g}(k) \leftarrow \mathbf{g}(k) / \ \mathbf{g}(k)\ $	r
$\mathbf{S}(k)_{(:,3)} = \mathbf{g}(k)$	
$\underline{\mathbf{C}}_{ys}(k)_{(:,3)} = \underline{\mathbf{C}}_{yy}(t)\mathbf{S}(k)_{(:,3)}$	r^2
$\mathbf{C}_{ss}(k)_{(:,3)} = \mathbf{S}(k)^H \underline{\mathbf{C}}_{ys}(k)_{(:,3)}$	$3r$
$\mathbf{C}_{ss}(k)_{(3,:)} = \mathbf{C}_{ss}(k)_{(:,3)}^H$	
Until $\ \nabla J(k)\ < \text{threshold}$	
$\lambda(t) \leftarrow \lambda(k)$	
$\phi(t) \leftarrow \mathbf{S}(k)_{(:,1)}$	

IV. SIMULATION RESULTS

In this section, the performance of YAST is illustrated and compared to that of other existing algorithms: the FAPI, LORAF3, and SP1 PST's, and the FDPM, HFRANS, and YAST-PGS MST's. Other performance comparisons, involving the same test signals, can be found in references [13], [15], [16],

⁷In particular, the simulation results presented in section IV are unchanged.

where the OPAST, QRI, NOOJA, and ODKA algorithms were compared to YAST. In the following simulations, the subspace trackers are initialized with the subspace weighting matrix formed by the r first columns of the $n \times n$ identity matrix.

A. Principal subspace tracking

In this section, the performance of the subspace estimation is analyzed in terms of the maximum principal angle between the true principal subspace of the correlation matrix $\mathbf{C}_{xx}(t)$ (obtained via an exact eigenvalue decomposition), and the estimated principal subspace of the same correlation matrix (obtained with the subspace tracker). This error criterion was initially proposed by P. Comon and G.H. Golub as a measure of the distance between equidimensional subspaces [20], and used as a PST performance factor in [6], [13], [21], [22]. The test signal is a sum of $r = 4$ complex sinusoidal sources plus a complex white Gaussian noise (the SNR is 5.7 dB). The frequencies of the sinusoids vary according to a jump scenario originally proposed by P. Strobach [21]: their values abruptly change at different time instants, between which they remain constant. Their variations are represented in Figure 1-a.

Figure 1-b shows the maximum principal angle error trajectory (averaged over $K = 50$ independent runs) obtained with several PST algorithms: FAPI [6], LORAF3 [21], SP1 [12], and YAST. Those algorithms were applied with the same vector length $n = 80$ and the same forgetting factor $\beta \approx 0.99$. It can be noticed that FAPI and LORAF3 behave similarly⁸. Besides, the results obtained with YAST and SP1 cannot be distinguished, which is not surprising, since those two algorithms compute the same subspace estimate (the computational complexity of YAST, however, is lower than that of SP1). It can be noticed that those two algorithms converge much faster than the other ones, and their principal angle error is always much lower. To the best of our knowledge, YAST and SP1 are the only PST algorithms which present such an exceptional convergence rate.

In order to measure the orthonormality of the estimated subspace weighting matrix, we calculate the same average performance factor as in [7]–[10]:

$$\eta(t) = \frac{1}{K} \sum_{k=1}^K \|\mathbf{W}_k(t)^H \mathbf{W}_k(t) - \mathbf{I}_r\|_F^2 \quad (43)$$

where the number of algorithm runs is $K = 50$, k indicates that the associated variable depends on the particular run, and $\|\cdot\|_F$ denotes the Frobenius norm. This orthogonality error is plotted in Figure 1-c. Computations were performed in double precision with Matlab[®] (64 bits precision). Aside the somewhat erratic behavior of SP1, the four algorithms prove to be stable. Besides, it can be noticed that YAST provides an improved orthonormality compared to SP1. The lowest orthonormality error is reached by FAPI.

⁸Contrary to what we erroneously suggested in the introduction of reference [6], the performance of LORAF3 in terms of subspace estimation also proves to be comparable to that of other classical implementations of the power iteration method, such as PAST [3], LORAF2 [4], OPAST [5], and NIC [23].

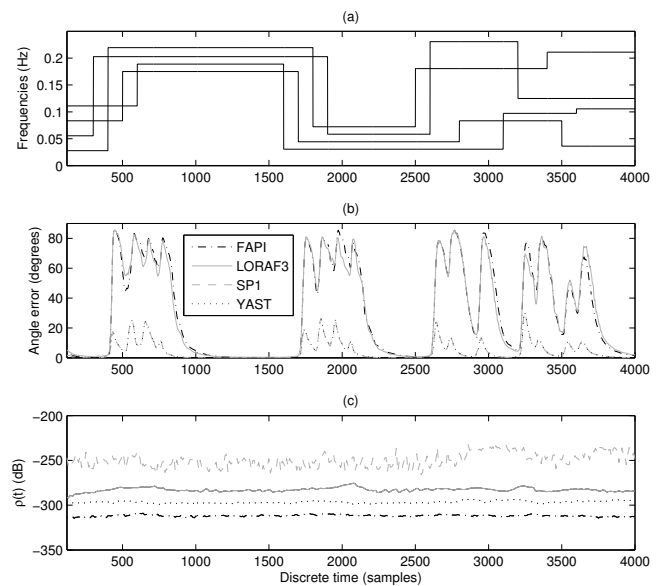


Fig. 1. Simulation results

- (a) Normalized frequencies of the sinusoids,
- (b) Angle error trajectories,
- (c) Departure from orthonormality.

B. Minor subspace tracking

1) *A classical example:* In the following, $\mathbf{x}(t)$ is a sequence of $n = 4$ dimensional independent jointly-Gaussian random vectors, with zero mean and covariance matrix

$$\mathbf{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix}.$$

This is a classical example, which served as a reference for testing several MST algorithms [7]–[10], with parameter $r = 2$. In order to measure the performance of the algorithms in terms of subspace estimation, we calculate the ensemble averages of the following performance factor, as in [7]–[10]:

$$\rho(t) = \frac{1}{K} \sum_{k=1}^K \frac{\text{trace}(\mathbf{W}_k(t)^H \mathbf{E}_1 \mathbf{E}_1^H \mathbf{W}_k(t))}{\text{trace}(\mathbf{W}_k(t)^H \mathbf{E}_2 \mathbf{E}_2^H \mathbf{W}_k(t))},$$

where the number of algorithm runs is $K = 50$, \mathbf{E}_1 is the exact $(n - r)$ -dimensional principal subspace, and \mathbf{E}_2 is the exact r -dimensional minor subspace of \mathbf{C} .

Figure 2-a shows the tracking results obtained with four MST's: our new implementation of YAST, the Pairwise Gram-Schmidt (PGS) implementation of YAST called YAST-PGS [18], and the HFRANS [8] and FDPM [2], [11] algorithms, which were derived from the Data Projection Method (DPM) of Yang and Kaveh [24]⁹. It can be noticed that the results obtained with HFRANS and FDPM cannot be

⁹The HFRANS and FDPM algorithms were implemented with parameter $\mu = 0.13$, and YAST with $\beta = 0.99$.

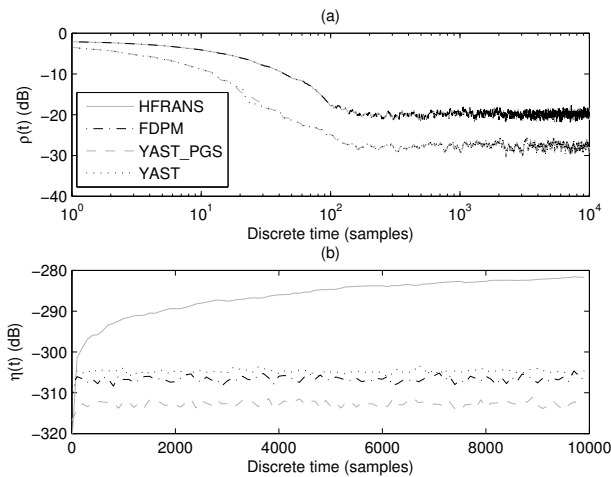


Fig. 2. A classical example
(a) Error trajectories,
(b) Departure from orthonormality.

distinguished, and that those obtained with YAST and YAST-PGS are almost identical. Moreover, both versions of YAST converge much faster than the algorithms derived from DPM, and the steady-state error obtained with YAST is lower. Figure 2-b shows the departures from orthogonality of the four above-mentioned subspace trackers, as defined in equation (43). It can be noticed that YAST, YAST-PGS and FDPM are stable, whereas the orthonormality error obtained with HFRANS keeps growing.

Note that this example, usually encountered in the literature, does not belong to the field of times series analysis. Therefore the optimization mentioned in section III-E cannot be applied, and the global complexity of YAST becomes n^2 in this case¹⁰.

2) *Time series analysis*: Here the YAST algorithm is applied to frequency estimation, so that its complexity is only $5nr$. The test signal is a sum of 4 complex sinusoidal sources, of same unitary amplitude, random phases, and normalized frequencies equal to 0.2, 0.4, 0.5 and 0.8. This signal is perturbed by an additive white Gaussian noise, so that the SNR is 30 dB. The data vectors are composed of $n = 12$ successive samples of the noisy signal, so that the dimension of the noise subspace is $r = 8$. However, the $n \times n$ matrix C is defined as the covariance matrix of the noiseless data vectors. The performance of the various subspace trackers is measured by the functions $\rho(t)$ and $\eta(t)$ defined above.

Figure 3-a shows the tracking results obtained with the four MST's, with the same parameters as above. The same observations can be made regarding the convergence properties of the various algorithms in terms of subspace estimation, except that our new implementation of YAST converges even faster than YAST-PGS. This is due to the negligible approximation introduced in section III-C, which surprisingly improves the

¹⁰Note that even in this unfavorable case, YAST remains competitive in comparison with the subspace trackers of linear complexity, since its much higher convergence rate reduces the total number of operations required for reaching a given error level.

convergence rate of YAST. Regarding the departures from orthogonality, represented in Figure 3-b, it can be noticed that the YAST-PGS algorithm slowly loses the orthogonality, whereas FDPM and our new implementation of YAST remain perfectly stable.

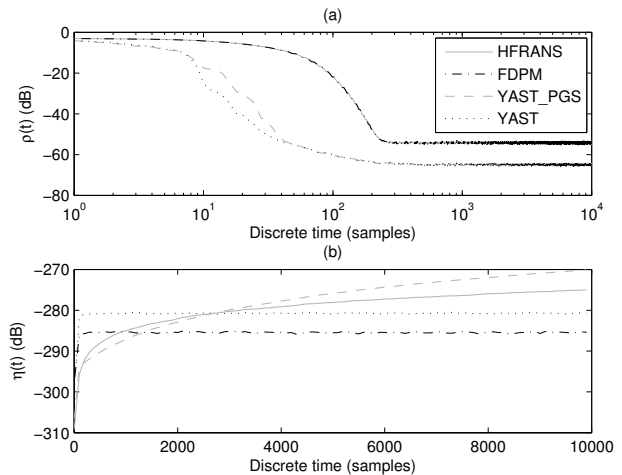


Fig. 3. Time series analysis
(a) Error trajectories,
(b) Departure from orthonormality.

V. CONCLUSIONS

In this paper, we presented a new implementation of the YAST algorithm for principal and minor subspace tracking, whose convergence and numerical stability are proved in Appendix B and Appendix C. The exceptional performance of YAST observed in our simulation results, both in terms of convergence rate and steady state error, is inherited from the SP1 algorithm by C.E. Davila [12]. Besides, the remarkable stability of this new implementation was obtained by applying an orthonormalization procedure similar to that of the FDPM algorithm by X.G. Doukopoulos and G.V. Moustakides [2], [11]. Regarding the computational complexity, if the data correlation matrix satisfies a shift invariance property, which typically holds in the case of time series analysis for instance, YAST involves only $5nr$ flops (instead of n^2 flops otherwise). This is to be compared to the $3nr$ complexity of the fastest PST algorithms, PAST [3] and FAPI [6], and the $4nr$ complexity of the fastest MST algorithms, among which NOOJA [10] and HFRANS [8].

For the clarity of the presentation, the proposed implementation involves a simple window shape for updating the correlation matrix, and a simple additional search direction at each iteration. Nevertheless, this implementation can be easily generalized to various window shapes and search directions, as proposed in [13]–[16].

APPENDIX

A. Asymptotic analysis

Proof of lemma III-C.1 (page 4): Substituting equations (2), (4), and (14) into equation (5) yields

$$\mathbf{\Pi}(t) - \mathbf{\Pi}(t-1) = \mathbf{u}(t)\mathbf{u}(t)^H - \mathbf{v}(t)\mathbf{v}(t)^H.$$

This equation can be interpreted as follows: the new subspace, represented by the projector $\mathbf{\Pi}(t)$, is obtained by adding the search direction $\mathbf{u}(t)$, and then removing the direction $\mathbf{v}(t)$ to the old subspace represented by $\mathbf{\Pi}(t-1)$. Then left and right multiplying this equation by $\mathbf{u}(t)$ and substituting equations (6), (14) and (24) yields

$$\mathbf{u}(t)^H (\mathbf{\Pi}(t) - \mathbf{\Pi}(t-1)) \mathbf{u}(t) = \varepsilon(t)^2.$$

Finally, since the matrix 2-norm is the norm induced by the Hermitian vector norm, we obtain $\varepsilon(t)^2 \leq \|\mathbf{\Pi}(t) - \mathbf{\Pi}(t-1)\|_2$. ■

Proof of lemma III-C.2 (page 4): Equations (6) and (25) yield

$$\mathbf{v}(t)^H \mathbf{T}(t) = \mathbf{\Phi}(t)^H \mathbf{W}(t)^H (\mathbf{W}(t-1) - \varepsilon(t)\mathbf{u}(t)\phi(t)^H). \quad (44)$$

Substituting equations (14) and (24) into equation (44) yields

$$\mathbf{v}(t)^H \mathbf{T}(t) = \varepsilon(t)(1 - \varphi(t))\theta(t)^* \phi(t)^H.$$

Noting that $1 - \varphi(t) = \frac{\varepsilon(t)^2}{1 + \varphi(t)}$, we finally obtain equation (26). ■

B. Convergence analysis

In this section we show that, assuming stationary data vectors of constant correlation matrix \mathbf{C}_{xx} , the YAST algorithm converges to the true principal or minor subspace of \mathbf{C}_{xx} . More precisely, assuming that $\forall t, \mathbf{C}_{xx}(t) = \mathbf{C}_{xx}$, and that the r -dimensional principal or minor subspace of \mathbf{C}_{xx} is unique¹¹, we show that the range space of the orthogonal projector $\mathbf{\Pi}(t) = \mathbf{W}(t)\mathbf{W}(t)^H$ converges to this subspace. In order to simplify the discussion, the following proof focuses on PST, but the same proof stands for MST. This proof is decomposed into 5 steps. We thus successively prove that:

- 1) there is a subsequence $\{\mathbf{\Pi}(t_n)\}$ of $\{\mathbf{\Pi}(t)\}$, which converges toward a rank- r orthogonal projector $\mathbf{\Pi}^*$;
- 2) $\mathbf{\Pi}^*$ is a critical point of the generalized Rayleigh quotient on the manifold \mathbb{P} of rank- r orthogonal projectors;
- 3) $\mathcal{J}(\mathbf{\Pi}(t))$ is an increasing convergent sequence;
- 4) $\mathbf{\Pi}^*$ is the global maximum of the generalized Rayleigh quotient on \mathbb{P} ;
- 5) $\mathcal{J}(\mathbf{\Pi}(t))$ converges to $\mathcal{J}(\mathbf{\Pi}^*)$;
- 6) the whole sequence $\mathbf{\Pi}(t)$ converges to $\mathbf{\Pi}^*$.

We first need to introduce the *natural gradient* $\nabla_{\mathbb{P}}\mathcal{J}$ of function \mathcal{J} on the manifold \mathbb{P} . By definition, $\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi})$ is the orthogonal projection of the gradient of function \mathcal{J} onto the tangent space of \mathbb{P} at point $\mathbf{\Pi}$. More precisely, it can be proved

¹¹In the case of PST, the r -dimensional principal subspace is uniquely defined if and only if $\lambda_r > \lambda_{r+1}$, where $\lambda_1 \geq \dots \geq \lambda_n \geq 0$ are the eigenvalues of \mathbf{C}_{xx} sorted in decreasing order.

that the natural gradient of the generalized Rayleigh quotient is¹²

$$\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}) = \mathbf{\Pi}\mathbf{C}_{xx} + \mathbf{C}_{xx}\mathbf{\Pi} - 2\mathbf{\Pi}\mathbf{C}_{xx}\mathbf{\Pi}. \quad (45)$$

This expression shows that the critical points of the generalized Rayleigh quotient (for which $\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}) = \mathbf{0}$) are the rank- r projectors which span an *invariant subspace* of \mathbf{C}_{xx} .

Step 1): Since \mathbb{P} is a compact set, the Bolzano-Weierstrass theorem proves that there is a subsequence $\{\mathbf{\Pi}(t_n)\}$ of $\{\mathbf{\Pi}(t)\}$, which converges toward a rank- r orthogonal projector $\mathbf{\Pi}^*$.

Step 2): Since $\mathbf{\Pi}(t)$ is the global maximum of the generalized Rayleigh quotient \mathcal{J} on the subset of orthogonal projectors whose range space is included in $\text{span}(\mathbf{\Pi}(t))$, $\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}(t)) \perp \mathbf{\Pi}(t)$. In particular, since $\mathbf{x}(t) \in \text{span}(\mathbf{\Pi}(t))$,

$$\forall t, \mathbf{x}(t) \perp \text{span}(\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}(t))). \quad (46)$$

Let $\varepsilon > 0$. Since \mathcal{J} is \mathcal{C}^1 , $\exists n_0$ such that $\forall n \geq n_0$ $\|\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}^*) - \nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}(t_n))\|_2 < \varepsilon$. Therefore $\forall n \geq n_0$, equation (46) yields

$$\begin{aligned} & \left\| \frac{\mathbf{x}(t_n)}{\|\mathbf{x}(t_n)\|_2}^H \nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}^*) \right\|_2 \\ &= \left\| \frac{\mathbf{x}(t_n)}{\|\mathbf{x}(t_n)\|_2}^H (\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}^*) - \nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}(t_n))) \right\|_2 \\ &\leq \|\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}^*) - \nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}(t_n))\|_2 < \varepsilon. \end{aligned}$$

However $\mathbf{x}(t_n)$ is a sequence of independent random vectors of full-rank covariance matrix; thus any family of n vectors in this sequence almost surely spans the whole space \mathbb{C}^n . This finally proves that $\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi}^*) = \mathbf{0}$, *i.e.* $\mathbf{\Pi}^*$ is a critical point of the generalized Rayleigh quotient on \mathbb{P} .

Step 3): Substituting equations (5), (4), (2), and (14) into equation (1) yields

$$\mathcal{J}(\mathbf{\Pi}(t)) = \mathcal{J}(\mathbf{\Pi}(t-1)) + \mathbf{u}(t)^H \mathbf{C}_{xx} \mathbf{u}(t) - \mathbf{v}(t)^H \mathbf{C}_{xx} \mathbf{v}(t).$$

In this equation, both unitary vectors $\mathbf{u}(t)$ and $\mathbf{v}(t)$ belong to the augmented subspace $\text{span}(\mathbf{\Pi}(t))$. However, $\mathbf{v}(t)$ corresponds the global minimum or the function $\mathbf{w} \mapsto \mathbf{w}^H \mathbf{C}_{xx} \mathbf{w} / \|\mathbf{w}\|_2^2$ within this subspace. Thus $\mathcal{J}(\mathbf{\Pi}(t))$ increases at each iteration. Moreover, since $\mathcal{J}(\mathbf{\Pi}(t))$ is bounded, $\mathcal{J}(\mathbf{\Pi}(t))$ is an increasing convergent sequence.

Step 4): Therefore the sequence $\mathbf{\Pi}(t_n)$, which converges to a critical point $\mathbf{\Pi}^*$ of \mathcal{J} on \mathbb{P} , is such that $\mathcal{J}(\mathbf{\Pi}(t_n))$ is increasing. However it is well known that the generalized Rayleigh quotient admits a unique local maximum on \mathbb{P} , which is the global maximum (other critical points are either unstable

¹²Let $\mathbb{T}_{\mathbf{\Pi}}$ denote the tangent space of the manifold \mathbb{P} at point $\mathbf{\Pi}$, and note that the standard gradient of function \mathcal{J} is the constant matrix \mathbf{C}_{xx} . By definition, $\nabla_{\mathbb{P}}\mathcal{J}(\mathbf{\Pi})$ is the orthogonal projection of \mathbf{C}_{xx} onto $\mathbb{T}_{\mathbf{\Pi}}$. Equation (45) thus follows from the following observations:

- $\mathbb{T}_{\mathbf{\Pi}}$ is the $(nr - r^2)$ -dimensional space of Hermitian matrices \mathbf{X} such that $\mathbf{\Pi}\mathbf{X} + \mathbf{X}\mathbf{\Pi} = \mathbf{X}$.
- $\mathbb{T}_{\mathbf{\Pi}}$ is also the linear space of all matrices of the form $\mathbf{\Pi}\mathbf{H} + \mathbf{H}\mathbf{\Pi} - 2\mathbf{\Pi}\mathbf{H}\mathbf{\Pi}$, where \mathbf{H} is Hermitian.
- the orthogonal complement $\mathbb{T}_{\mathbf{\Pi}}^{\perp}$ of $\mathbb{T}_{\mathbf{\Pi}}$ is the $(n^2 + r^2 - nr)$ -dimensional space of all matrices of the form $\mathbf{A} + \mathbf{H} - \mathbf{\Pi}\mathbf{H} - \mathbf{H}\mathbf{\Pi} + 2\mathbf{\Pi}\mathbf{H}\mathbf{\Pi}$, where \mathbf{A} is skew-Hermitian and \mathbf{H} is Hermitian.
- Let \mathbf{M} be any $n \times n$ matrix, and let \mathbf{A} denote its skew-Hermitian part and \mathbf{H} denote its Hermitian part. Then the orthogonal projection of \mathbf{M} onto $\mathbb{T}_{\mathbf{\Pi}}$ is $\mathbf{\Pi}\mathbf{H} + \mathbf{H}\mathbf{\Pi} - 2\mathbf{\Pi}\mathbf{H}\mathbf{\Pi}$, and the orthogonal projection of \mathbf{M} onto $\mathbb{T}_{\mathbf{\Pi}}^{\perp}$ is $\mathbf{A} + \mathbf{H} - \mathbf{\Pi}\mathbf{H} - \mathbf{H}\mathbf{\Pi} + 2\mathbf{\Pi}\mathbf{H}\mathbf{\Pi}$.

saddle points, or the global minimum). Thus $\mathbf{\Pi}^*$ is the unique global maximum of the generalized Rayleigh quotient on \mathbb{P} .

Step 5): Since $\mathcal{J}(\mathbf{\Pi}(t_n))$ converges to the maximum value \mathcal{J}^* of the generalized Rayleigh quotient and the whole sequence $\mathcal{J}(\mathbf{\Pi}(t))$ is increasing, $\mathcal{J}(\mathbf{\Pi}(t))$ also converges to \mathcal{J}^* .

Step 6): Finally, let prove by contradiction that the whole sequence $\mathbf{\Pi}(t)$ converges to $\mathbf{\Pi}^*$. Suppose that $\exists \varepsilon > 0$ and a subsequence $\mathbf{\Pi}(t_k)$ of $\mathbf{\Pi}(t)$ such that $\forall k, \|\mathbf{\Pi}(t_k) - \mathbf{\Pi}^*\|_2 \geq \varepsilon$. Let \mathbb{P}_ε be the set of all rank- r orthogonal projectors $\mathbf{\Pi}$ such that $\|\mathbf{\Pi} - \mathbf{\Pi}^*\|_2 \geq \varepsilon$, and let $\mathcal{J}_\varepsilon^*$ be the supremum of \mathcal{J} on \mathbb{P}_ε . Since \mathcal{J} is a continuous function, and \mathbb{P}_ε is a compact set, the extreme value theorem proves that \mathcal{J} attains its supremum $\mathcal{J}_\varepsilon^*$ on \mathbb{P}_ε . However $\mathbf{\Pi}^*$ is the unique global maximum of \mathcal{J} on \mathbb{P} ; thus $\mathcal{J}^* > \mathcal{J}_\varepsilon^*$. Consequently, the subsequence $\mathbf{\Pi}(t_k)$ satisfies $\forall k, \mathcal{J}(\mathbf{\Pi}(t_k)) \leq \mathcal{J}_\varepsilon^* < \mathcal{J}^*$. This contradicts the fact that $\mathcal{J}(\mathbf{\Pi}(t))$ converges to \mathcal{J}^* . As a conclusion, the whole sequence $\mathbf{\Pi}(t)$ converges to $\mathbf{\Pi}^*$.

C. Numerical stability

As mentioned in the introduction, our previous implementations of YAST [13], [15] suffered from numerical stability problems when the algorithm had converged, as observed in [17] in the case of PST, and as proved in [18] in the case of MST. In this section, we prove that our new implementation is stable both for PST and MST. Indeed, suppose that, because of rounding errors, $\mathbf{W}(t-1)^H \mathbf{W}(t-1) = \mathbf{I}_r + \mathbf{E}(t-1)$ where $\mathbf{E}(t-1) \neq \mathbf{0}$, and let show that the orthogonality error $\mathbf{E}(t)$ induced at time t is smaller than that at time $t-1$. In the following developments, we first show that matrix $\mathbf{E}(t)$ satisfies a simple recursion. Then, noting that this recursion fits into the framework of the proof presented in [2], we refer to this proof to conclude about the numerical stability of YAST.

First, let $\mathbf{e}(t) = -\mathbf{W}(t-1)^H \mathbf{u}(t)$. Equations (10) and (13) show that $\mathbf{e}(t) = \frac{\mathbf{E}(t-1)\mathbf{y}(t)}{\sigma(t)}$. Then matrix $\mathbf{T}(t)$ defined in equation (25) satisfies

$$\begin{aligned} \mathbf{T}(t)^H \mathbf{T}(t) &= \mathbf{I}_r + \varepsilon(t)^2 \phi(t) \phi(t)^H + \mathbf{E}(t-1) \\ &\quad + \varepsilon(t) (\mathbf{e}(t) \phi(t)^H + \phi(t) \mathbf{e}(t)^H) \\ &= \mathbf{I}_r + \mathcal{O}(\varepsilon(t)^2) + \mathbf{E}(t-1) + \mathcal{O}(\varepsilon(t) \|\mathbf{E}(t-1)\|_2). \end{aligned}$$

As mentioned in section III-C, when the algorithm has reached its steady state, $\varepsilon(t) \ll 1$. Thus the previous equation can be simplified in the form $\mathbf{T}(t)^H \mathbf{T}(t) \approx \mathbf{I}_r + \mathbf{E}(t-1)$, where \approx stands for the approximation $\varepsilon(t) \ll 1$. Then equation (31) yields $\mathbf{Q}(t)^H \mathbf{Q}(t) \approx \mathbf{I}_r + \mathbf{S}(t) \mathbf{E}(t-1) \mathbf{S}(t)$. Since matrix $\mathbf{W}(t)$ is obtained by normalizing the first column of $\mathbf{Q}(t)$, this last equation shows that matrix $\mathbf{D}(t)$ in equation (32) is of the form $\mathbf{D}(t) = \mathbf{I}_r - \frac{\varepsilon(t)}{2} \mathbf{e}_1 \mathbf{e}_1^T$, where $\varepsilon(t) = \mathcal{O}(\|\mathbf{E}(t-1)\|_2)$ and $\mathbf{e}_1 = [1, 0, \dots, 0]^T$. Finally, equation (32) yields $\mathbf{W}(t)^H \mathbf{W}(t) \approx \mathbf{I}_r + \mathbf{E}(t) + \mathcal{O}(\|\mathbf{E}(t)\|_2^2)$, where

$$\mathbf{E}(t) = \mathbf{S}(t) \mathbf{E}(t-1) \mathbf{S}(t) - \varepsilon(t) \mathbf{e}_1 \mathbf{e}_1^T. \quad (47)$$

However it can be noticed that the top left coefficient of $\mathbf{E}(t)$ is necessary zero since the first column of $\mathbf{W}(t)$ is explicitly normalized. Thus $\varepsilon(t)$ is also the top left coefficient of matrix $\mathbf{S}(t) \mathbf{E}(t-1) \mathbf{S}(t)$, i.e. $\varepsilon(t) = \mathbf{e}_1^T \mathbf{S}(t) \mathbf{E}(t-1) \mathbf{S}(t) \mathbf{e}_1$.

By considering the column-wise version $\text{col}\{\mathbf{E}(t)\}$ of the error $\mathbf{E}(t)$ (i.e. where the matrix elements are read in a column by column manner), equation (47) can be rewritten in the form

$$\text{col}\{\mathbf{E}(t)\} = \mathbf{P}_{r,2} (\mathbf{S}(t) \otimes \mathbf{S}(t)) \text{col}\{\mathbf{E}(t-1)\} \quad (48)$$

where \otimes denotes the Kronecker product, and $\mathbf{P}_{r,2} = \mathbf{I}_{r,2} - (\mathbf{e}_1 \otimes \mathbf{e}_1)(\mathbf{e}_1^T \otimes \mathbf{e}_1^T)$. Based on an equation similar to equation (48), detailed mathematical developments proving the numerical stability of FDPMM were provided in Appendix III in [2], in the framework of stochastic approximation theory. These developments can also be applied here, to finally prove the stability of the YAST algorithm. The basic idea is the following: the algorithm is stable if and only if the eigenvalues of $\mathbf{P}_{r,2} (\mathbf{S}(t) \otimes \mathbf{S}(t))$ are strictly smaller than one. This is almost surely the case because $\mathbf{P}_{r,2}$ is a rank deficient orthogonal projector (which generally lowers the vector norm), and $\mathbf{S}(t) \otimes \mathbf{S}(t)$ is a random unitary matrix (which maintains the vector norm).

REFERENCES

- [1] A.-J. Van der Veen, E. F. Deprettere, and A. L. Swindlehurst, "Subspace based signal analysis using singular value decomposition," *Proc. of IEEE*, vol. 81, no. 9, pp. 1277–1308, Sept. 1993.
- [2] X. Doukopoulos and G. Moustakides, "Fast and Stable Subspace Tracking," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1452–1465, Apr. 2008.
- [3] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. Signal Processing*, vol. 44, no. 1, pp. 95–107, Jan. 1995.
- [4] P. Strobach, "Low-rank adaptive filters," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 2932–2947, Dec. 1996.
- [5] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Proc. Letters*, vol. 7, no. 3, pp. 60–62, Mar. 2000.
- [6] R. Badeau, B. David, and G. Richard, "Fast Approximated Power Iteration Subspace Tracking," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2931–294, Aug. 2005.
- [7] P. Strobach, "Square root QR inverse iteration for tracking the minor subspace," *IEEE Trans. Signal Processing*, vol. 48, no. 11, pp. 2994–2999, 2000.
- [8] S. Attallah, "The Generalized Rayleigh's Quotient Adaptive Noise Subspace Algorithm: A Householder Transformation-Based Implementation," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 1, pp. 3–7, Jan. 2006.
- [9] S. C. Douglas, "Numerically-robust adaptive subspace tracking using Householder transformations," in *Proc. of Sensor Array and Multichannel Signal Proc. Workshop*. IEEE, 2000, pp. 499–503.
- [10] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Proc. Letters*, vol. 8, no. 7, pp. 203–206, 2001.
- [11] X. Doukopoulos and G. Moustakides, "Blind Adaptive Channel Estimation in OFDM Systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1716–1725, July 2006.
- [12] C. E. Davila, "Efficient, high performance, subspace tracking for time-domain data," *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp. 3307–3315, Dec. 2000.
- [13] R. Badeau, B. David, and G. Richard, "Yet Another Subspace Tracker," in *Proc. of 2005 International Conference on Acoustics, Speech, and Signal Processing ICASSP'05*, vol. 4. IEEE, Mar. 2005, pp. 329–332.
- [14] R. Badeau, "Méthodes à haute résolution pour l'estimation et le suivi de sinusoides modulées. Application aux signaux de musique." Ph.D. dissertation, École Nationale Supérieure des Télécommunications, Paris, Apr. 2005, in French.
- [15] R. Badeau, B. David, and G. Richard, "YAST Algorithm for Minor Subspace Tracking," in *Proc. of 2006 International Conference on Acoustics, Speech, and Signal Processing ICASSP'06*, vol. 3. Toulouse, France: IEEE, May 2006, pp. 552–555.
- [16] —, "Conjugate gradient algorithms for minor subspace analysis," in *2007 International Conference on Acoustics, Speech, and Signal Processing ICASSP'07*, vol. 3, Honolulu, Hawaii, USA, Apr. 2007, pp. 1013–1016.

- [17] J. Altuna, B. Mulgrew, R. Badeau, and V. Atxa, "A fast adaptive method for subspace based blind channel estimation," in *Proc. of 2006 International Conference on Acoustics, Speech, and Signal Processing ICASSP'06*, vol. 4. Toulouse, France: IEEE, May 2006, pp. 1121–1124.
- [18] S. Bartelmaos and K. Abed-Meraim, "An efficient & stable algorithm for minor subspace tracking and stability analysis," in *Proc. of ICASSP'07*, vol. III. Honolulu, Hawaii, USA: IEEE, Apr. 2007, pp. 1301–1304.
- [19] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927–935, Nov./Dec. 1992.
- [20] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore and London: The Johns Hopkins University Press, 1996.
- [21] P. Strobach, "Fast recursive subspace adaptive ESPRIT algorithms," *IEEE Trans. Signal Processing*, vol. 46, no. 9, pp. 2413–2430, Sept. 1998.
- [22] R. Badeau, G. Richard, and B. David, "Sliding window adaptive SVD algorithms," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 1–10, Jan. 2004.
- [23] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Processing*, vol. 46, no. 7, pp. 1967–1979, July 1998.
- [24] J. Yang and M. Kaveh, "Adaptive Eigensubspace Algorithms for Direction or Frequency Estimation and Tracking," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 2, pp. 241–251, Feb. 1988.