ELSEVIER

Signal Processing 74 (1999) 253–277

SIGNAL PROCESSING

On the efficient use of Givens rotations in SVD-based subspace tracking algorithms

Philippe Pango*, Benoît Champagne

INRS-Télécommunications, Université du Québec, 16 place du Commerce, Verdun, Québec, Canada H3E 1H6

Received 28 July 1997; received in revised form 23 October 1998

Abstract

In this paper, the issue of the efficient use of Givens rotations in SVD-based QR Jacobi-type subspace tracking algorithms is addressed. By relaxing the constraint of upper triangularity on the singular value matrix, we show how even fewer Givens rotations can achieve a better diagonalization and provide more accurate singular values. Then, we investigate the efficient use of Givens rotations as a vector rotation tool. The cancellation of cross-terms is presented as an efficient signal/noise separation technique which guarantees a better updating of the subspaces basis. Regarding the choice between inner and outer rotations, we properly use the permutation properties of Givens rotations to maintain the decreasing ordering of the singular values throughout the updating process and analyze the consequences on the tracking performance of QR Jacobi-type algorithms. Finally, based on the developed theory, we propose two efficient subspace tracking algorithms which outperform existing QR Jacobi-type algorithms. Comparative simulation experiments validate the concepts.

Zusammenfassung

In diesem Beitrag wird die effiziente Verwendung von Givens rotations in SVD-basierten subspace tracking Algorithmen vom QR Jacobi-Typ diskutiert. Wir zeigen, dass durch die Aufgabe der Einschränkung auf obere Dreiecksstruktur der Singulärwertmatrix mit weniger Givens rotations eine bessere Diagonalisierung and genauere Singulärwerte erzielt werden können. Weiters wird die effiziente Verwendung von Givens rotations als ein vector rotation Werkzeug diskutiert. Die gegenseitige Aufhebung von Kreuztermen wird als effizientes Signal/Geraüsch Trennungsverfahren präsentiert, welches ein besseres updating der Unterraumbasis garantiert. Wir demonstrieren den Einfluss der Rotationsart (innere oder aüssere) auf die Reihenfolge der Singulärwerte während des gesamten Aktualisierungsprozesses. Weiters werden die Konsequenzen der Rotationsart auf die tracking performance diskutiert. Schliesslich werden zwei effiziente subspace tracking Algorithmen vorgeschlagen, die auf den gewonnenen Ergebnissen basieren und besser sind als existierende Algorithmen vom QR Jacobi-Typ. Vergleichende Simulationsergebnisse bestätigen die vorgestellen Konzepte.

Résumé

Dans cet article, nous abordons le problème de l'utilisation optimale des rotations de Givens dans les algorithmes de type QR Jacobi pour le suivi de sous-espaces. En enlevant la contrainte de triangularité supérieure, nous montrons

* Corresponding author. Tel.: 514-761-8635; fax: 514-761-8501; e-mail: pango@inrs-telecom.uquebec.ca

0165-1684/99/$ – see front matter
PII: S0165-1684(98)00215-1

comment un nombre réduit de rotations de Givens rotations peuvent produire une meilleure diagonalisation de la matrice des valeurs singulières et donner des valeurs singulières plus précises. Puis, nous analysons l'utilisation efficace des rotations de Givens en tant qu'outils de rotation de vecteurs. L'annulation des termes croisés est présentée comme une technique efficace de séparation des sous-espaces signal et bruit qui garanti une meilleure mise à jour des bases des sous-espaces. À propos du choix entre rotations internes et externes, nous exploitons adéquatement les propriétés permutatives de ces rotations pour maintenir l'ordre décroissant des valeurs singulières lors de la mise à jour, et analysons les conséquences sur les performances de suivi des algorithmes de type QR Jacobi. Finalement, en se basant sur la théorie développée, nous proposons deux algorithmes efficaces de suivi de sous-espaces dont les performances excèdent celles des algorithmes de type QR Jacobi existant. Des simulation expérimentales valident les concepts proposés.

Keywords: Subspace tracking; Invariant subspace updating; Singular-value decomposition; Givens rotation; Cross-terms; Direction of arrival

Notation

a	angle of arrival
A	data matrix
ΔV_S	time variation of signal subspace
Δ_l	convergence step induced by the lth rotation
γ	efficiency of Givens rotations (EGR)
$G_\theta^{i\|j}$	Givens rotation in i–j plane with angle θ
k	time index
λ	forgetting factor
M	number of pairs of rotations at refinement step
N	number of sensors
off[ ]	off-norm of its argument
r	number of sources
σ_i	ith singular value
$\bar{\sigma}_N$	average noise singular value
Σ	singular value matrix
U	left singular vectors
V	right singular vectors
ω	electrical angle
$\boldsymbol{x}$	measurement vector
$\boldsymbol{y}$	update vector, i.e. projection of $\boldsymbol{x}$ on V
$\tilde{\boldsymbol{y}}_i$	update vector after the ith QR rotation

1. Introduction

In the application of subspace methods to array signal processing, projection techniques like MUSIC [25], root-MUSIC [1], ESPRIT [24] or minimum-norm [15] are often used to estimate the signal parameters. These methods require a subspace information which can be provided by various decompositions, including the eigenvalue decomposition (EVD) of the sample correlation matrix, the QR factorization [2] or the singular-value decomposition (SVD) of the data matrix. In the case of non-stationary signals, these decompositions need to be updated each time new information is provided by the most recent measurements. The quality of the updating process affects the accuracy of the parameter estimates, and its low complexity may allow a real-time implementation. These considerations have led to an intensive research on the development of the so-called subspace tracking algorithms. Beginning with Owsley [18], many adaptive subspace tracking algorithms have been developed and can be grouped in families, depending on the specific technique they use: stochastic gradient [35], recursive least squares [34], perturbation approach [3], to quote a few. The introduction in [5] is a good source of references concerning the development of subspace tracking algorithms.

Generally, an SVD-based subspace tracking consists in computing the SVD of a data matrix of growing dimension, defined recursively as

$$A(k) = \begin{bmatrix} \sqrt{\lambda} A(k-1) \\ \boldsymbol{x}^{\mathrm{H}}(k) \end{bmatrix}, \tag{1}$$

where k is the discrete time index, $0 < \lambda < 1$ is the forgetting factor, and $\boldsymbol{x}(k) \in \mathbb{C}^N$ is the measurement vector at time index k. The SVD of $A(k)$ can be expressed as

$$A(k) = U(k)\Sigma(k)V(k)^{\mathrm{H}}, \tag{2}$$

where $V(k)$ is an $N \times N$ unitary matrix, $U(k)$ is a $k \times N$ matrix with orthonormal columns and $\Sigma(k)$ is an $N \times N$ diagonal matrix. Usually, the left singular vectors $U(k)$ need not be tracked to provide the subspace information. So, one is interested in tracking only the singular values $\{\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_N\}$ along the diagonal of $\Sigma(k)$ and the right singular vectors, i.e. the columns of $V(k)$. Computing this SVD from scratch at each iteration is computationally expensive and time-consuming. The main issue consists then in using the information contained in the measurement $\boldsymbol{x}(k)$ to update the previous decomposition obtained at time $k-1$.

In this work, we focus our attention on a specific family of SVD-based subspace tracking algorithms which we define as QR Jacobi-type algorithms [12,17,23]. QR Jacobi-type algorithms intensively make use of Givens rotations in the subspace updating process, the main matrix operations consisting in products with Givens rotations; thus, they track the subspaces with a low complexity and a rather simple formulation. Also, Givens rotations have the advantage to maintain the orthonormality of matrices, assuming an infinite precision. The direct consequence is that QR Jacobi-type algorithms are good candidates for implementation in a rather regular structure, e.g. using CORDIC processors [27,32].

While the exact SVD-based tracking of $V(k)$ and $\Sigma(k)$ requires $O(N^3)$ operations at each iteration, Moonen et al. have proposed a QR Jacobi-type algorithm with a complexity of $O(N^2)$ operations per update [17]. In this paper, we shall refer to Moonen's algorithm as MA. Kavcic and Yang [12] introduced a noise sphericalized version of MA, the noise average SVD (NASVD) which updates a sphericalized URV decomposition [30]; it reduces the complexity to $O(Nr)$ by tracking only an r-dimensional signal subspace. Based on the same structure, Rabideau [23] proposed an algorithm of similar complexity, the refinement-only fast subspace tracking (RO-FST), which considerably improves the tracking performance. Let us note that Stewart's algorithm for the updating of the rank revealing URV [30] is also a QR Jacobi-type algorithm which has the advantage to track the number of sources r.

The effectiveness of the above algorithms in subspace tracking and/or parameter estimation has already been validated through simulations. Yet many questions remain unanswered regarding these algorithms. Since they use exclusively Givens rotations throughout the updating process, one must analyze the efficiency of these rotations on the singular values and singular vectors tracking performance, and also on the accuracy of the signal parameter estimates they provide.

Regarding the singular values, QR Jacobi-type algorithms usually track an approximate SVD decomposition (e.g. URV) in which the singular value matrix $\Sigma(k)$ is given a specific structure; then, Givens rotations are applied throughout the updating algorithm in order not to destroy that specific structure. It might be argued that such a structural constraint introduces limitations in the way Givens rotations could be used to achieve a better updating of the singular values. For example, the above QR Jacobi-type algorithms impose $\Sigma(k)$ in Eq. (2) to be upper-triangular. The efficiency of Givens rotations assuming this specific structure (and eventually others) has not been investigated yet. Therefore, one notes that the above algorithms use Givens rotations in a way which is not proved to be optimal.

Regarding the singular vectors, QR Jacobi-type algorithms use Givens rotations to rotate the approximate singular vectors in the direction of the exact ones. This re-orientation of the singular vectors is performed to update a subspace basis which inputs MUSIC-like estimators and provide the desired signal parameter estimates. The accuracy of the parameter estimates is linked to the ability of the Givens rotations to rotate the proper singular vectors in the proper direction. Here, the issue consists then in the choice of the appropriate vectors to be rotated in order to provide the best update of the subspace basis. Note that in most existing QR Jacobi-type algorithms, this issue is not specifically addressed.

Finally, a general issue is the ordering of the singular values throughout the updating process. It is generally accepted that the ordering of the singular values can be controlled by the choice of the type of rotations: *inner* or *outer*. It is also known that the choice of the type of rotations has a major effect on the tracking performance of QR Jacobi-type algorithms; the analytical demonstration of the link between the type of rotations and both the ordering of the singular values and the tracking

error of QR Jacobi-type algorithms needs to be clearly stated.

The purpose of this paper is to investigate the efficient use of Givens rotations in order to propose new efficient SVD-based subspace tracking algorithms. To diagonalize $\Sigma(k)$ efficiently, we remove the constraint of upper triangularity of $\Sigma(k)$ and propose an updating structure in which the Givens rotations are dynamically applied in various subregions of $\Sigma(k)$, allowing a robust response to highly non-stationary cases. We analyze the direct effect of Givens rotation on the subspace basis tracking error and introduce the concept of cross-terms cancellation as an efficient signal/noise subspace separation technique. We provide a solution to the question of the ordering of the singular values by choosing appropriately the type of each Givens rotation (inner or outer). Our conclusions regarding the structure of the decomposition, the type of Givens rotations and their efficient use enable us to validate the relative performance of various existing QR Jacobi-type algorithms and propose two new computationally efficient QR Jacobi-type subspace tracking algorithms. They are formulated for both cases of partial and complete tracking, i.e signal + noise subspace tracking and signal subspace-only tracking, with respective complexity $O(N^2)$ and $O(Nr)$, where r is the number of signal sources. The new algorithms outperform existing QR Jacobi-type algorithms, even with fewer Givens rotations.

The paper is organized as follows. In Section 2, the efficient use of Givens rotations to diagonalize $\Sigma(k)$ and track the subspaces basis vectors is addressed; also, the choice between inner or outer rotations and its effect on the tracking error and the ordering of the singular values is analyzed. Section 3 presents the proposed algorithms, and Section 4 shows experimental results of computer experiments to validate the new concepts and algorithms in various tracking scenarios. Finally, conclusive remarks are given in Section 5. Throughout the paper, $\|\cdot\|$ is the 2-norm of its argument.

2. Optimizing the use of Givens rotation

Refering to Eq. (2), the convergence of an SVD-based subspace tracker to the exact SVD of $A(k)$ is closely related to two matters: the ability of the algorithm to diagonalize $\Sigma(k)$, and then, to rotate the approximate singular vectors in the direction of the exact ones. Indeed, a good diagonalization of $\Sigma(k)$ ensures the correctness of the singular values, while an appropriate rotation of the estimated singular vectors moves them close to the direction of the exact ones. The Givens rotation is a tool that can achieve both tasks simultaneously if it is judiciously used. In this section, we first present some preliminaries after which we discuss the issue of the efficient use of Givens rotations for the tracking of the singular values; then, we do the same for the singular vectors and, finally, we investigate the type of rotations to be used, i.e. inner or outer, and its consequence on the tracking performance.

2.1. Preliminaries

Let us first introduce the appropriate notation for the Givens rotation, which is the main tool QR Jacobi-type algorithms use. We recall that a Givens rotation differs from an identity matrix only at four entries as in

$$G_\theta^{i|j} = \begin{bmatrix} I_{i-1} & & & & \\ & c & \cdots & s^* & \\ & \vdots & I_{j-i-1} & \vdots & \\ & -s & \cdots & c & \\ & & & & I_{N-j} \end{bmatrix}, \tag{3}$$

where I_n is the $n \times n$ identity matrix and $|c|^2 + |s|^2 = 1$. In the real case, we have $c = \cos\theta$ and $s = \sin\theta$, where θ is the angle of the rotation. Note that a complex Givens rotation G_c can be expressed as the product of a real Givens rotation G_r with a phasor P, as in $G_c = G_r P$; accordingly in this paper, we may use the real notation without any loss of generality.

Referring to Eqs. (1) and (2), one would like to approximate the decomposition of the data matrix $A(k)$. To update the singular-value matrix Σ and the right singular vectors V from time $k-1$ to k, QR Jacobi-type subspace tracking algorithms essentially proceed in two steps, namely the *QR step* and the *refinement step*. The QR step applies N Givens

rotations to zero each entry of the measurement vector's projection on the observation space, i.e. $\boldsymbol{y}(k) = V(k-1)^{\mathrm{H}}\boldsymbol{x}(k)$; the updated matrix $\Sigma(k)$ is obtained by dropping the last (zeroed) row:

$$\begin{bmatrix} \Sigma(k) \\ \mathbf{0}^{\mathrm{H}} \end{bmatrix} = [G_{\phi_N}^{N|N+1}]^{\mathrm{H}} \cdots [G_{\phi_2}^{2|N+1}]^{\mathrm{H}}[G_{\phi_1}^{1|N+1}]^{\mathrm{H}} \times \begin{bmatrix} \sqrt{\lambda}\Sigma(k-1) \\ \boldsymbol{y}(k)^{\mathrm{H}} \end{bmatrix}. \tag{4}$$

Regarding the update vector $\boldsymbol{y}(k)$, note furthermore that one may either pre-multiply each right singular vectors by an appropriate phase term to make $\boldsymbol{y}(k)$ real [26] and maintain a real singular value matrix $\Sigma(k)$, or work directly with complex Givens rotations. In the latter case, phase terms have to be interleaved with Givens rotations to maintain the singular values real along the diagonal of $\Sigma(k)$. In the sequel, one may assume that the components of $\boldsymbol{y}(k)$ and $\Sigma(k)$ are real without any loss of generality. After the QR step, the *refinement step* intends to concentrate the energy of $\Sigma(k)$ along its diagonal. This 'diagonalization' step, sometimes called *SVD step* or *refinement*, exists in different flavours in the literature. The refinement usually consists in a series of M pairs of Givens rotations on both sides of $\Sigma(k)$ [12,17], as in

$$\Sigma(k) \leftarrow [G_{\phi_l}^{i_l|j_l}]^{\mathrm{H}}\Sigma(k)G_{\theta_l}^{i_l|j_l} \tag{5}$$

for $l = 1:M$, and the corresponding accumulation of the right rotations on the singular vectors

$$V(k) \leftarrow V(k)\prod_{l=1}^{M} G_{\theta_l}^{i_l|j_l}. \tag{6}$$

In Eq. (5), the rotation angles θ_l and ϕ_l are computed so as to zero the (i_l, j_l) and (j_l, i_l) entries of $\Sigma(k)$, which we denote as $\sigma_{i_l j_l}$ and $\sigma_{j_l i_l}$; we call $\sigma_{i_l j_l}$ the *pivot* of the rotations $G_{\phi_l}^{i_l|j_l}$ and $G_{\theta_l}^{i_l|j_l}$. For each (i_l, j_l) pair of entries to be zeroed at the refinement step, the angles ϕ_l and θ_l are the solutions of the SVD (diagonalization) of a 2-by-2 matrix, as in

$$\begin{bmatrix} \cos\phi_l & \sin\phi_l \\ -\sin\phi_l & \cos\phi_l \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \sigma_{i_l i_l} & \sigma_{i_l j_l} \\ \sigma_{j_l i_l} & \sigma_{j_l j_l} \end{bmatrix} \begin{bmatrix} \cos\theta_l & \sin\theta_l \\ -\sin\theta_l & \cos\theta_l \end{bmatrix} = \begin{bmatrix} \sigma'_{i_l i_l} & 0 \\ 0 & \sigma'_{j_l j_l} \end{bmatrix}. \tag{7}$$

Details for the computation of θ_l and ϕ_l can be found in [16]. Note that the rotations can be either *inner* or *outer*. Later in this paper, we shall address the issue of the choice of the type of rotations.

In MA, we have $M = N-1$ so that the overall complexity of the updating algorithm is $O(N^2)$. Kavcic's NASVD [12] is the sphericalized version of MA. It lowers the updating complexity by tracking the following decomposition:

$$A(k-1) = U(k-1)\begin{bmatrix} \tilde{\Sigma}(k-1) & \mathbf{0} \\ \mathbf{0} & \bar{\sigma}_N(k-1)I_{N-r-1} \end{bmatrix} \times V(k-1)^{\mathrm{H}}. \tag{8}$$

$\tilde{\Sigma}(k-1)$ is an upper-triangular square matrix of dimension $r+1$; its diagonal contains the r signal singular values and one average noise singular value $\bar{\sigma}_N(k-1)$ which is kept at the $(r+1, r+1)$ position. The above decomposition is updated in four steps: a Householder transformation which rotates the noise subspace basis so that the projection of the incoming data $\boldsymbol{x}(k)$ on it has a single non-zero component [11], a QR step and a refinement step not different from Eqs. (4)–(6), but requiring fewer Givens rotations, i.e. $M = r$. Finally, the average noise singular value is re-estimated. The complexity of NASVD is $O(Nr)$.

2.2. A matter of diagonalization

In this section, we would like to develop updating algorithms which diagonalize the singular value matrix $\Sigma(k)$ efficiently, i.e. with M not exceeding $N-1$ pairs of Givens rotations at the refinement step as in MA. In this case, the overall complexity of the algorithm is expected not to be higher than $O(N^2)$. Below, we show that the solution to the efficient diagonalization of $\Sigma(k)$ consists in choosing appropriately the (i_l, j_l) pivots in Eq. (5).

2.2.1. An appropriate structure for $\Sigma(k)$

In QR Jacobi-type algorithms, the main goal of the refinement step is the reduction of the off-norm of $\Sigma(k)$ which is defined by

$$\mathrm{off}[\Sigma(k)] = \sqrt{\sum_{1 \le i \ne j \le N} |\sigma_{ij}|^2}. \tag{9}$$

Let us reconsider the refinement step as described in Eq. (5) and omit temporarily the rotation index l. After the pair of rotations $[G_{\phi}^{i|j}]^{\mathrm{H}}$ and $G_{\theta}^{i|j}$ are applied to the left and the right of $\Sigma(k)$, respectively, its off-norm is reduced as [9,10]

$$\mathrm{off}[\Sigma(k)]^2 \leftarrow \mathrm{off}[\Sigma(k)]^2 - (|\sigma_{ij}|^2 + |\sigma_{ji}|^2). \tag{10}$$

Thus, the larger σ_{ij} and σ_{ji}, the better the diagonalization. This observation led us to investigate the effect of selecting the off-diagonal entry of $\Sigma(k)$ with the largest magnitude before each pair of rotations is applied, so that in Eq. (5) the rotations always zero the largest pivots. We call this technique a 'Maximum Search' (MS) procedure. In agreement with Eq. (10) and as observed experimentally in Section 4, such a technique achieves a better off-norm reduction, and thus a better tracking of the singular values.

Note that unlike other QR Jacobi-type algorithms, the MS procedure does not maintain the upper-triangular structure. Indeed, since off-diagonal maxima can be located anywhere in $\Sigma(k)$, the MS procedure applies Givens rotations anywhere and not necessarily on the second diagonal of $\Sigma(k)$ as in conventional QR Jacobi-type algorithms. This causes non-zero upper elements to be reintroduced in the lower part of $\Sigma(k)$, breaking the upper-triangular structure. To reduce efficiently the off-norm by allowing Givens rotations to be applied anywhere in $\Sigma(k)$, we need to consider a different type of approximate decomposition, still expressed as in (2), but where $\Sigma(k)$ is now almost diagonal but not specifically upper-triangular. We refer to this decomposition as an UXV, where X stands for the singular-value matrix $\Sigma(k)$ which has no particular structure but being almost diagonal. As the URV [30], the UXV is simply an approximate SVD but in which the constraint of upper triangularity of Σ has been relaxed to provide more flexibility to Givens rotations.

As far as the SVD is concerned, the most important attribute for $\Sigma(k)$ is not to be upper-triangular as required by conventional QR Jacobi-type algorithms, but to have the smallest possible off-norm. On one hand, applying rotations as conventional QR Jacobi-type algorithms do, i.e. preserving the upper triangular structure, is not proved to be the optimal way to reduce the off-norm of $\Sigma(k)$ under the constraint of a fixed number of Givens rotations. On the other hand, the breaking of the upper-triangular structure does not interfere with the main goal of the refinement step: the reduction of the off-norm of $\Sigma(k)$. Note also that unlike the full matrix, an upper-triangular $\Sigma(k)$ does not allow one to restore the decreasing ordering of the singular values at any time, even by using permutations.

As a consequence of the UXV decomposition, the MS technique does not guarantee the standard QR in Eq. (4) to zero totally the incoming vector $\boldsymbol{y}(k)$ in a fixed number of Givens rotations. More particularly, the ith QR rotation $G_{\phi_i}^{i|N+1}$ in Eq. (4) zeros the ith entry of $\boldsymbol{y}(k)$ and modifies only the ith and the $(N+1)$th rows, as in

$$[i\text{th row}] \leftarrow \cos\phi_i[i\text{th row}] - \sin\phi_i[(N+1)\text{th row}], \tag{11}$$

$$[(N+1)\text{th row}] \leftarrow \cos\phi_i[(N+1)\text{th row}] + \sin\phi_i[i\text{th row}]. \tag{12}$$

As a consequence of Eq. (12) being repeated N times for $i = 1, \ldots, N$ in Eq. (4), it can be verified that after the QR step, the entries of the last row are of the same order of magnitude as the off-diagonal entries of $\Sigma(k)$. This implies that after the QR rotations have been applied, the diagonalization error due to the entries of the $(N+1)$th row is much smaller than the total error due to all the off-diagonal entries of $\Sigma(k)$, assuming N is reasonably large. Also, one should notice that in practical applications, the data matrix itself $A(k)$ is noise-corrupted; therefore, depending on the SNR and the forgetting factor λ, the effect of setting to 0 ε-size terms (where ε is the order of magnitude of the off-diagonal entries of $\Sigma(k)$) might be masked by the statistical fluctuations resulting from the noise.

Based on the above considerations, as an updating procedure for the UXV decomposition, we propose to drop the entries of the last row once the N QR Givens rotations have been applied. Even though this UXV-based QR step does not zero completely $\boldsymbol{y}(k)^{\mathrm{H}}$ and does not lead to an information-preserving updating procedure, we still refer to such an operation as a QR step to maintain a compatibility of terminology with existing algorithms.

In summary, assuming an UXV decomposition, an MS refinement appears as an efficient off-norm

reduction procedure. Our viewpoint is that what is lost at the QR step is meaningless, if compared to the improvement of performance due to the 'maximum-search' diagonalization procedure. The removal of the upper-triangular constraint of $\Sigma(k)$ leads effectively to improved subspace tracking algorithms, as observed experimentally in Section 4.

2.2.2. Locating efficiently the largest pivots

At the refinement step, the location of the largest off-diagonal entry of $\Sigma(k)$ cannot be determined a priori from theoretical consideration alone. Indeed, after the QR step, the largest off-diagonal entry of $\Sigma(k)$ can be located anywhere, and its position changes within $\Sigma(k)$ after each pair of rotation is applied. The only way to obtain the maximum location is by an exhaustive search. Locating the largest entry in an $N \times N$ array involves N^2 comparison tests. Since the MS procedure requires the maximum search to be performed prior to each of the M pairs of rotations at the refinement step, this leads to an unacceptable $O(N^3)$ complexity updating algorithm. (Recall that M is $O(N)$.) In fact, even if all the entries of $\Sigma(k)$ are maintained in a separate sorted table, each refinement rotation in Eq. (5) modifies $8N-4$ entries (2 rows and 2 columns) which have to be re-inserted in the table; this still leads to an $O(N^2 \log N)$-complexity bottleneck. Therefore, one needs to find a way to perform the maximum search, but at a lower complexity.

To efficiently reduce the off-norm of $\Sigma(k)$, the key idea is to predict the sub-region of $\Sigma(k)$ where the off-norm is expected to increase more significantly after the QR step; if the search region contains $O(N)$ entries, e.g. a row, so an MS-like procedure can be performed locally with a reduced complexity. The main issue then is the accurate detection of the sub-region of $\Sigma(k)$ where the QR step causes the largest norm increase. In this paper, we propose an approach which consists in estimating the off-norm increase in each row of $\Sigma(k)$ after the QR step and distribute dynamically the refinement rotations between all the rows.

If $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_N$ are the singular values of an $M \times N$ matrix B listed in decreasing order, one defines the effective rank of B as r if $\sigma_r > \varepsilon > \sigma_{r+1}$, where $1 \leqslant r \leqslant \min(M,N)$ and ε is a specified bound [14]. Let r denote the effective rank of $A(k)$. We define the following submatrices:

$$U(k) = [U_S(k)|U_N(k)], \qquad V(k) = [V_S(k)|V_N(k)], \tag{13}$$

where $U_S(k) = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r] \in \mathbb{C}^{k \times r}$ and $V_S(k) = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r] \in \mathbb{C}^{N \times r}$. Let us also rewrite the almost diagonal singular-value matrix as

$$\Sigma(k) = \begin{bmatrix} \Sigma_S(k) & \Sigma_{SN}(k) \\ \Sigma_{NS}(k) & \Sigma_N(k) \end{bmatrix}, \tag{14}$$

where $\Sigma_S(k) \in \mathbb{R}^{r \times r}$, and the other blocks are defined accordingly.

The main diagonal of $\Sigma(k)$ contains the approximate singular values which for now are supposed to be always maintained in decreasing order: $\sigma_1 > \sigma_2 > \cdots > \sigma_N$; later in the paper, we shall address the issue of the ordering of the singular values. In Eqs. (13) and (14), the singular values and singular vectors have thus been separated into two sets corresponding to the r largest and $N-r$ smallest singular values; the signal and noise subspaces are the column-span of $V_S(k)$ and $V_N(k)$, respectively. Let $\boldsymbol{y}(k) = V(k-1)^{\mathrm{H}}\boldsymbol{x}(k) = [\boldsymbol{y}_S(k)|\boldsymbol{y}_N(k)]$; $\boldsymbol{y}_S(k)$ and $\boldsymbol{y}_N(k)$ represent the components of the projection of the measurement vector $\boldsymbol{x}(k)$ on the signal and noise subspaces, respectively.

One easily notes that the size of the entries of $\boldsymbol{y}_S$ and $\boldsymbol{y}_N$ provides a draft estimation of the localization of the off-norm increase in Σ. Indeed, at the QR step the norm $\|\boldsymbol{y}_S(k)\|$ is transferred exclusively to Σ_S, while $\|\boldsymbol{y}_N(k)\|$ is transferred to Σ_{SN} and Σ_N. This information might be usefull, it is still not complete yet. Searching for the largest entry in Σ_{SN} remains a complex operation, particularly when $N \gg r$. To reduce the search region to the size of a row, let us investigate more thoroughly the off-norm increase of $\Sigma(k)$ at the QR step.

We assume that the off-diagonal entries of $\Sigma(k)$ are originally ε-sized before the QR step where ε is small. Let us denote by $\tilde{\boldsymbol{y}}_i$ the update vector $\boldsymbol{y}(k)$ altered by the i first consecutive rotations of the QR step, with $\tilde{\boldsymbol{y}}_0 = \boldsymbol{y}(k)$. In Eq. (4), the ith QR rotation $[G_{\phi_i}^{i|N+1}]^{\mathrm{H}}$ alters only the ith and the $(N+1)$th row, i.e. $\tilde{\boldsymbol{y}}_{i-1}^{\mathrm{H}}$, as in Eqs. (11) and (12) which we now

rewrite as

$$[i\text{th row}] \leftarrow [i\text{th row}]\cos\phi_i - \tilde{\boldsymbol{y}}_{i-1}^{\mathrm{H}}\sin\ \phi_i, \quad (15)$$

$$\tilde{\boldsymbol{y}}_i^{\mathrm{H}} \leftarrow \tilde{\boldsymbol{y}}_{i-1}^{\mathrm{H}}\cos\phi_i + [i\text{th row}]\sin\ \phi_i. \quad (16)$$

On the one hand, the angle ϕ_i is computed so as to zero the ith entry of the last row $\tilde{\boldsymbol{y}}_{i-1}^{\mathrm{H}}$; one sees from Eq. (16) that $[G_{\phi_i}^{i|N+1}]^{\mathrm{H}}$ also alters the entire $(N+1)$th row by adding to it a vector whose entries are the off diagonal ε-sized entries from the ith row, multiplied by $\sin\ \phi_i$. On the other hand, one sees from Eq. (15) that $[G_{\phi_i}^{i|N+1}]^{\mathrm{H}}$ alters the ith row by multiplying it by $\cos\phi_i$ and adding $\tilde{\boldsymbol{y}}_{i-1}^{\mathrm{H}}\sin\ \phi_i$ to it. Therefore, in $\Sigma(k)$, the off-norm growth (ng) generated by the ith QR rotation is located in the ith row only, and can be approximated by the norm of the term which is transferred from the $(N+1)$th row to the off-diagonal entries of the ith row, i.e.

$$\mathrm{ng}(i) = \|\tilde{\boldsymbol{y}}_{i-1}(1\cdots i-1,\ 0,\ i+1\cdots N)\sin\ \phi_i\|. \quad (17)$$

The computation of the off-norm increase $\mathrm{ng}(i)$ of each row of $\Sigma(k)$ prior to the refinement step can be used advantageously. Indeed, the refinement step has to be performed with only M pairs of Givens rotations; instead of searching for the maximum of an N-by-N array prior to each pair of rotations, one can dynamically distribute the M refinement pairs of rotations between the $N-1$ rows of $\Sigma(k)$ according to the predicted off-norm growth $\mathrm{ng}(i)$ in each of them, i.e. more pivots are annihilated in the rows whose off-norm increase more significantly. Once the number of Givens rotations allocated to a specific row is known, e.g. $l(i)$, an $\mathrm{O}(N)$-complexity maximum search can be performed $l(i)$ times on the ith row only prior to each of the $l(i)$ refinement pairs of Givens rotations. Thus, if M is of the order of N as in MA, the maximum search has a low complexity of $\mathrm{O}(N^2)$.

In this paper, we propose a solution to the efficient reduction of the off-norm of $\Sigma(k)$ which combines the above row and block-division approach of $\Sigma(k)$.

The idea is to distribute the available M pairs of Givens rotations to each row section in each block. The whole procedure is presented in Table 1; it consists in interleaving the computation of the off-norm increase in each row section with the QR step. The dynamic distribution of the rotations is performed in two steps:

1. *Estimate the off-norm growth in each row section for each block*: this is performed at the QR step by computing $\mathrm{ng}_{SN}(i)$, $\mathrm{ng}_N(i)$ and $\mathrm{ng}_N(i)$ as in Table 1, prior to each of the $N-1$ first QR rotation; here, the ng's correspond to the norm of the term which is transferred from the $(N+1)$th row to each row section in each block.
2. *In each block, distribute the Givens rotations to each row section*: we compute the total off-norm increase

$$\mathrm{NG} = \sqrt{\sum_{i=1}^{r}\mathrm{ng}_S^2(i) + \sum_{i=1}^{r}\mathrm{ng}_{SN}^2(i) + \sum_{i=r+1}^{N}\mathrm{ng}_N^2(i)} \quad (18)$$

then, we allocate to each row section a number of pairs of Givens rotations corresponding to its contribution to NG.

In Σ_S, we compute the number of pairs of rotations to be applied in each row section, $l_S(i)$ for $i = 1:r$. Similarly, in Σ_{SN}, we compute $l_{SN}(i)$ for $i = 1{:}r$ and finally, the number of rotations in the row sections of Σ_N, $l_N(i)$ for $i = r+1{:}N$, so that we have $M = \sum_{i=1}^{r} l_S(i) + \sum_{i=1}^{r} l_{SN}(i) + \sum_{i=r+1}^{N} l_N(i)$. Once each row section has been allocated a specific number of Givens rotations, the maximum search can be performed prior to each rotation, but in the corresponding row section only. This procedure leads to a low complexity maximum-search refinement and, as will be shown in simulation experiments, to an efficient diagonalization of $\Sigma(k)$ even with fewer Givens rotations.

2.3. A matter of vector rotation

The above discussion has been drawn from a 'singular-value' point of view. On a diagonalization point of view, it is advantageous to choose the largest off-norm entries as pivots, as this ensures an efficient reduction of the off-norm of $\Sigma(k)$. But, as shown below, it does not guarantee that the singular vectors are updated properly. In this section, we now turn our attention on the effect of Givens rotations on the estimated singular vectors only.

The QR Givens rotations consist in row rotations on $\Sigma(k)$ and have no effect at all on the right singular vectors in $V(k)$. After the QR step, the

Table 1
QR step and off-norm increase computation

Step	Operation
QR step and off-norm growth estimation for each row section	for $i = 1{:}N$ OFF-NORM GROWTH ESTIMATION compute the angle ϕ_i to zero $\boldsymbol{y}(i)$ as in $[G_{\phi_i}^{1\|2}]^{\mathrm{H}}\begin{bmatrix}\sigma_{ii}\\ \boldsymbol{y}(i)\end{bmatrix}=\begin{bmatrix}\sigma'_{ii}\\ 0\end{bmatrix}$ if $i \leqslant r$ $\mathrm{ng}_S(i) = \|\boldsymbol{y}(1\cdots i-1,0,i+1\cdots r)\sin\phi_i\|$ $\mathrm{ng}_{SN}(i) = \|\boldsymbol{y}(r+1{:}N)\sin\phi_i\|$ else $\mathrm{ng}_N(i) = \|\boldsymbol{y}(r+1\cdots i-1,0,i+1\cdots N)\sin\phi_i\|$ end QR ROTATION $\begin{bmatrix}\Sigma\\ \boldsymbol{y}^{\mathrm{H}}\end{bmatrix}\leftarrow[G_{\phi_i}^{i\|N+1}]^{\mathrm{H}}\begin{bmatrix}\Sigma\\ \boldsymbol{y}^{\mathrm{H}}\end{bmatrix}$ end
Compute the number of pairs of rotations in each row section of each block	$NG=\sqrt{\sum_{i=1}^{r}ng_S^2(i)+\sum_{i=1}^{r}ng_{SN}^2(i)+\sum_{i=r+1}^{N}ng_N^2(i)}$ for $i = 1{:}N-1$ if $i \leqslant r$ $l_S(i) = \mathrm{round}(M\mathrm{ng}_S(i)/\mathrm{NG})$ $l_{SN}(i) = \mathrm{round}(M\mathrm{ng}_{SN}(i)/\mathrm{NG})$ else $l_N(i) = \mathrm{round}(M\mathrm{ng}_N(i)/\mathrm{NG})$ end end
Maintain the number of rotations in Σ_{SN} greater or equal to 1	if $\sum_{i=1}^{r} l_{SN}(i) = 0$ find the row section of Σ_{SN} with the largest norm-increase i.e. find h such that $l_{SN}(h) = \max([l_{SN}(1)\cdots l_{SN}(r)])$ set $l_{SN}(h) = 1$ end

intermediate decomposition has the form of Eq. (2) and needs to be further refined. At this point, we have

$$A(k) = U_S(k)\Sigma_S(k)V_S(k)^{\mathrm{H}} + U_N(k)\Sigma_N(k)V_N(k)^{\mathrm{H}} + A_{SN}(k), \tag{19}$$

where

$$A_{SN}(k) = U_S(k)\Sigma_{SN}(k)V_N(k)^{\mathrm{H}} + U_N(k)\Sigma_{NS}(k)V_S(k)^{\mathrm{H}}. \tag{20}$$

$A_{SN}(k)$ expresses the cross-interaction between subspaces of different nature (e.g. between the signal right singular vectors and the noise left singular vectors). Let us express the exact decomposition at time k as

$$A(k) = U_S^{\text{`}\circ\text{'}}(k)\Sigma_S^{\text{`}\circ\text{'}}(k)V_S^{\text{`}\circ\text{'}}(k)^{\mathrm{H}} + U_N^{\text{`}\circ\text{'}}(k)\Sigma_N^{\text{`}\circ\text{'}}(k)V_N^{\text{`}\circ\text{'}}(k)^{\mathrm{H}}, \tag{21}$$

where '$\circ$' stands for 'exact decomposition'. $\Sigma_S^{\text{`}\circ\text{'}}$ and $\Sigma_N^{\text{`}\circ\text{'}}$ are $r\times r$ and $(N-r)\times(N-r)$ diagonal submatrices, containing the exact signal and noise singular values.

By comparing Eqs. (19) and (21), we see that the approximate expression is more likely to be close to the one of an SVD when $A_{SN}(k) = \mathbf{0}$, which is achieved when the off-diagonal blocks in Eq. (14) are null, i.e. $\|\Sigma_{SN}\| = \|\Sigma_{NS}\| = 0$. We note that even when $A_{SN}(k) = \mathbf{0}$, the approximate subspaces are not yet identical to the exact ones since the

submatrices Σ_S and Σ_N may not be diagonal; nevertheless, the block diagonalization of $\Sigma(k)$ helps to improve the separation between the signal and noise subspaces.

To substantiate this affirmation, we will now analyze the direct effect of the above-mentioned block diagonalization on the tracking error. More specifically, we shall investigate the effect of Givens rotations on the signal subspace projector $V_S(k)V_S(k)^H$. Recall that this projector is used by MUSIC-like algorithms to estimate various signal parameters. Let the tracking error be defined as the distance between the approximate and exact signal subspace projector, i.e.

$$\mathrm{TE}(k) = \| V_S^\circ(k)V_S^\circ(k)^H - V_S(k)V_S(k)^H \|. \tag{22}$$

Similarly, define the *time variation* of the updating process [17], $\Delta V_S(k)$, as the distance between the approximate signal subspaces at two consecutive time iterations, $k-1$ and k, i.e.

$$\Delta V_S(k) = \| V_S(k)V_S(k)^H - V_S(k-1)V_S(k-1)^H \|. \tag{23}$$

In QR Jacobi-type subspace tracking algorithms, the refinement step transforms the right singular vectors according to Eq. (6) where (i_l, j_l) represents the pivot location and θ_l the angle of the lth right Givens rotation in Eq. (5). Each right rotation in Eq. (6) affects only the i_lth and the j_lth right singular vectors as in

$$\begin{aligned} \boldsymbol{v}_{i_l} &\leftarrow \boldsymbol{v}_{i_l}\cos\theta_l - \boldsymbol{v}_{j_l}\sin\theta_l, \\ \boldsymbol{v}_{j_l} &\leftarrow \boldsymbol{v}_{i_l}\sin\theta_l + \boldsymbol{v}_{j_l}\cos\theta_l. \end{aligned} \tag{24}$$

Temporarily denote by $U(l-1)\Sigma(l-1)V(l-1)^H$ the approximate decomposition of $A(k)$ before the lth rotation is applied. In the same way, define $V_S(l-1)$, $V_N(l-1)$, $\Sigma_S(l-1)$, $\Sigma_N(l-1)$, $\Sigma_{SN}(l-1)$ and $\Sigma_{NS}(l-1)$ as in Eqs. (14) and (13) with time index k replaced by rotation index $l-1$. Then, we may define Δ_l, a measure of the partial alteration of the signal subspace projector resulting from the lth rotation, as the distance between the approximated signal subspaces before and after that rotation in Eq. (6), i.e.

$$\Delta_l = \| V_S(l)V_S(l)^H - V_S(l-1)V_S(l-1)^H \|. \tag{25}$$

Whereas $\Delta V_S(k) = 0$ means either perfect convergence or a useless algorithm at the kth time iteration, $\Delta_l = 0$ is the result of an unnecessary Givens rotation. Since each rotation in Eq. (5) helps to improve the diagonalization of $\Sigma(k)$ and refine the decomposition, we intuitively consider the time variation $\Delta V_S(k)$ as a convergence step, while Δ_l can be viewed as a measure of the contribution of the lth rotation to $\Delta V_S(k)$. Our viewpoint here is that, like an MS algorithm which performs a better diagonalization at each rotation by zeroing the off-diagonal entry with the largest magnitude, Δ_l should be as large as possible.

After the lth pair of rotations is applied, the modification of the signal subspace is expressed as

$$V_S(l) = V_S(l-1)G^{ul} + V_N(l-1)G^{ll}, \tag{26}$$

where G^{ul} and G^{ll} denote, respectively, the $r \times r$ *upper-left* and $(N-r)\times r$ *lower-left* part of $G_{\theta_l}^{i_l|j_l}$. There are three possible choices for the location of the lth pivot in $\Sigma(l-1)$, namely: $\Sigma_S(l-1)$, $\Sigma_N(l-1)$ and $\Sigma_{SN}(l-1)$ ($\Sigma_{SN}(l-1)$ and $\Sigma_{NS}(l-1)$ being equivalent). For each of these locations, let us analyze the effect of the lth Givens rotation:

1. $r+1 \leqslant i_l < j_l \leqslant N$: the pivot is located in $\Sigma_N(l-1)$. We have $G^{ul} = I_r$ and $G^{ll} = \mathbf{0}$; the rotation affects only two noise singular vectors as in Eq. (24). Since the signal subspace $V_S(l-1)$ is unchanged, we have $\Delta_l = 0$.
2. $1 \leqslant i_l < j_l \leqslant r$: the pivot is located in $\Sigma_S(l-1)$ and the rotation affects two vectors of the signal subspace. Here, $G^{ll} = \mathbf{0}$ and G^{ul} is unitary so that, using Eqs. (25) and (26), one still obtains $\Delta_l = 0$, even though two signal vectors are rotated.
3. $1 \leqslant i_l \leqslant r$ and $r+1 \leqslant j_l \leqslant N$: the pivot is located in $\Sigma_{SN}(l-1)$. In this case, one vector is rotated in each subspace. The signal subspace transformation is still expressed as in Eq. (26), but here, $G^{ll} \neq \mathbf{0}$ and G^{ul} is not unitary. As a result, $\Delta_l > 0$ and a positive contribution is made to the time variation $\Delta V_S(k)$.

In summary, during the refinement step, each rotation $G_{\theta_l}^{i_l|j_l}$ modifies the signal subspace projector $V_S(k)V_S(k)^H$ and thus contributes to the time variation $\Delta V_S(k)$ if and only if the pivot is chosen in Σ_{SN}, or similarly, if the pair of rotations rotate two singular vectors belonging respectively to the

orthogonal subspaces spanned by V_S and V_N. Then, it follows that locating the pivots in Σ_{SN} guarantees that each Givens rotation produces a noticeable effect on typical detectors such as MUSIC, and therefore on the estimated parameters; otherwise, the parameters estimates remain the same. Several papers from Stewart [28] and Fiero et al. [7,8] justify the need to reduce $\|\Sigma_{SN}\|$ so as to minimize an upper bound on the tracking error, assuming either an URV decomposition [30] or an ULV decomposition [31]. Indeed, each rotation in Σ_{SN} reduces $\|\Sigma_{SN}\|$ by annihilating the cross-terms σ_{i,j_i} and $\sigma_{j_i i_r}$.

We propose to measure the 'Efficiency of Givens rotations' (EGR) of a Jacobi-type algorithm as the proportion of cross-terms rotations involved in its refinement step, as in

$$\gamma(N,r)=\mathrm{Avg}\left[\frac{\text{number of cross-terms } 2\times 2 \text{ SVDs}}{\text{number of } 2\times 2 \text{ SVDs}}\right], \tag{27}$$

where Avg[ ] stands for time average. In our algorithm design philosophy, we consider that the EGR should be high enough to guarantee a satisfactory updating of the signal subspace projector, and accordingly, a better subspace tracking performance, without increasing the total number of rotations.

2.4. '*Inner*' or '*Outer*'

The conclusions of the previous section were obtained under the assumption that the cross-terms are confined in the upper right $r\times(N-r)$ corner of $\Sigma(k)$. To easily identify the cross-terms region of $\Sigma(k)$, one must be able to monitor and maintain the decreasing ordering of the singular values throughout the updating process. Note that once the ordering of the singular values is correct, no permutations are needed to restore the appropriate ordering of the singular values and vectors. The ordering of the singular values mainly depends on two factors: the initialization of the process, and the decision to choose inner or outer rotations. In this section, we discuss the proper choice of the type of rotations in Jacobi-type algorithms so as to maintain the ordering of the singular values and provide a higher EGR.

Let us reconsider the refinement of QR Jacobi-type algorithms shown in Eq. (5) in which we temporarily abandon the rotation index l. A series of pairs of rotations $[G_\theta^{i|j}]^{\mathrm{H}}$ and $G_\phi^{i|j}$ are used to diagonalize $\Sigma(k)$. Each of these pairs performs the SVD of a 2-by-2 matrix and refine the singular values σ_{ii} and σ_{jj} into σ'_{ii} and σ'_{jj}, as in Eq. (7). Regarding the angles θ and ϕ, there exist two solutions which zero the off-diagonal terms σ_{ij} and σ_{ji}, namely inner or outer rotation angles [16]. To force the updated singular values σ'_{ii} and σ'_{jj} to be in decreasing ordering, i.e. $\sigma'_{ii}>\sigma'_{jj}$, it can be proved that the type of the rotations in must be chosen as follows [19]:

$$\sigma_{ii}^2 \underset{\text{outer}}{\overset{\text{inner}}{\gtrless}} \sigma_{jj}^2+\sigma_{ij}^2-\sigma_{ji}^2. \tag{28}$$

Eq. (28) illustrates the mixing properties of outer and inner rotations. This simple test guides one in the choice between inner and outer rotations to obtain two updated singular values σ'_{ii} and σ'_{jj} in decreasing ordering. When the off-diagonal entries are much smaller than the diagonal entries, we generally have $\sigma_{ii}^2>\sigma_{jj}^2+\sigma_{ij}^2-\sigma_{ji}^2$; in this case, outer rotations generally permute the singular values unlike inner rotations. Note that those mixing properties of outer rotations had already been noticed by Stewart [29]; in [19], we provide the analytical proof of the sorting effect of inner and outer rotations.

The main conclusion is that the ordering of the singular values can be imposed by the choice of the type of rotations according to the magnitude of the involved off-diagonal and diagonal entries. Before each pair of rotations in Eq. (5), one can easily choose the type of the rotations to maintain the ordering of the corresponding pair of singular values. This 2-by-2 ordering might not lead to a perfect ordering of *all* the singular values; but, it generally ensures that the signal singular values and vectors are maintained at the top r positions. In this case, the cross-terms entries, i.e. $\Sigma_{SN}(k)$, are succesfully confined in the $r\times(N-r)$ upper-right region of $\Sigma(k)$, and eventually a full cross-terms annihilation refinement can be performed.

To elaborate on the above conclusion, let us analyze here how various QR Jacobi-type

algorithms select the type of rotations, and the corresponding effect on their convergence properties. In the algorithm from Moonen et al. (MA) [17], all $N-1$ rotations are *outer* and in Eq. (5), $j_l = i_l + 1$ for $i_l = 1, \ldots, N-1$; the pivots are chosen along the second diagonal of $\Sigma(k)$, from (1,2) down to $(N-1,N)$. In agreement with Eq. (28), if the off-diagonal entries are sufficiently small, this type of sweep provides a built-in permutation scheme which is due to the permutation properties of outer rotations, as demonstrated in [19]. The result is that in Eq. (5), the first singular value (not necessarily the largest) moves from one position to the next one after each pair of rotations is applied, so that at the end of the sweep, it reaches the last position. The EGR of MA averaged over N samples is

$$\gamma_{\mathrm{ma}}(N,r) = \frac{2r(N-r)}{N(N-1)}. \tag{29}$$

In fact, during the first r time iterations, we have $\gamma_{\mathrm{ma}} = (N-r)/(N-1)$; then γ_{ma} falls to $r/(N-1)$ during the following $N-r$ time iterations. Recall that the EGR indicates the proportion of pair of rotations which updates the signal subspace projector. Accordingly, in each window of N samples long, the tracking performance of MA can be expected to be higher during the r first time iterations, and lower during the $(N-r)$ following samples, assuming N is reasonably large; this pattern is then repeated every N samples, when the decreasing ordering of the singular values is recovered once again (see simulations in Section 4). Fig. 1 shows how γ_{ma} varies as a function of N for different values of r. One sees that in general, 50% of the rotations do not alter the subspaces projection.

In NASVD, as in any other algorithm which sphericalizes the noise subspace, it is crucial to lock the average noise singular value $\bar{\sigma}_N$ at the last position, otherwise a signal singular value might be

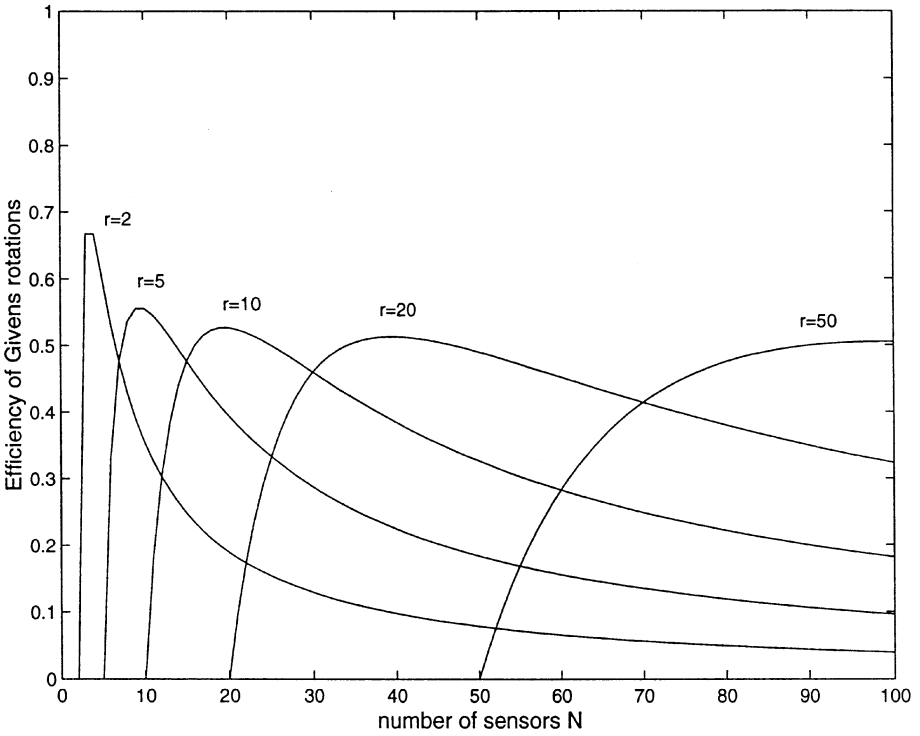

Fig. 1. Efficiency of Givens rotation in Moonen's algorithm (MA) as a function of N and r.

wrongly averaged with the noise singular values. Therefore, Kavcic proposes a refinement step in which all rotations are outer except the last one. Let us first note that by choosing the last rotation to be an inner one, NASVD becomes highly dependent on the initialization: indeed, the singular values must be selected so that the initial average noise singular value is small enough, since it has few chances to be permuted to an upper position. Also, since $\bar{\sigma}_N$ is locked at the $(r+1)$th position, the only rotation which anihilates a cross-term in each sweep, i.e. corresponding to Σ_{SN}, is the last rth one, so that the EGR of NASVD over r samples is only

$$\gamma_{\text{nasvd}} = 1/r. \tag{30}$$

In the case of sources tracking, e.g. direction of arrival (DOA) tracking, NASVD cannot be expected to perform well when two sources get close. Indeed, it is known that when two sources get much closer, one signal singular value falls to the noise level. Even if this singular value becomes the smallest of the set, the exclusive use of outer rotations in the $(r-1)$ first rotations hardly brings it to the $(r+1)$th position. For all these reasons, the tracking performance of NASVD cannot be expected to be high, as will be shown in simulations.

The last QR Jacobi-type algorithm we investigate is a modified version of NASVD, Refinement Only Fast Subspace Tracking (ROFST) from Rabideau [23]. Essentially, ROFST differs from NASVD by its refinement structure which zeros the last column and the last row of $\tilde{\Sigma}(k-1)$ alternately with two separate series of row and column rotations. The two separate series of Givens rotations implement inner rotations which are more likely to maintain the ordering of the singular values if the off-diagonal entries are sufficiently small. By choosing all the pivots in the last column, ROFST has the highest possible EGR, $\gamma_{\text{fst}} = 1$. This explains the good performance of ROFST as compared to NASVD and other subspace tracking algorithms, as observed in [22].

3. Proposed algorithms

The discussion in Section 2 leads to the following principles regarding the use of Givens rotations in QR Jacobi-type algorithms:

- an UXV decomposition enables more flexibility in the choice of pivots locations;
- the computation of the off-norm increase of $\Sigma(k)$ at the QR step enables one to distribute dynamically the Givens rotations and perform a better diagonalization;
- by choosing appropriately the type of rotations, one can maintain the larger singular values at the top r diagonal positions;
- at each time iteration, at least one rotation in Eq. (5) must be applied to annihilate cross-terms in $\Sigma(k)$ to ensure the updating of the subspace projection matrix.

In this section, we take into account these principles in the design of two new efficient subspace tracking algorithms for both cases of partial and complete tracking, i.e. signal + noise subspace tracking and signal subspace-only tracking.

3.1. Signal and noise subspace tracking: cross-terms singular-value decomposition 2 (CSVD2)

Based on the theoretical considerations of the previous Section, we propose an improved second version of the CSVD algorithm [20], namely CSVD2. It retains the cross-terms annihilation concept of CSVD, but eventually, allocates a portion of the refinement Givens rotations to $\Sigma_N(k)$ and $\Sigma_S(k)$ to provide a better and robust tracking of the singular values and the singular vectors. The CSVD2 algorithm is presented in Table 2 and its steps are explained below.

Initialization: $V(0) = I_N$ and $\Sigma(0) = \mathbf{0}_N$, or with an initial approximate SVD.

QR step and off-norm increase computation (see Table 1): the QR step is achieved by N row rotations as in Eq. (4). For each of these rotations, e.g. the ith rotation $[G_{\phi_i}^{i|N+1}]^{\mathrm{H}}$, we first compute the angle ϕ_i; then we compute the off-norm increase in the row sections of the only row of $\Sigma(k)$ that is altered by $[G_{\phi_i}^{i|N+1}]^{\mathrm{H}}$, i.e. the ith row. Finally, the rotation is applied so as to zero the ith entry of the last row. At the end of the QR step, the entries of the last row are assumed small enough to be dropped.

Distribution of the refinement Givens rotations (see Table 1): the M pairs of refinement rotations are

Table 2
CSVD2 algorithm

Step	Operation
Initialization	Initialize with $V = I_N$ and $\Sigma = \mathbf{0}_N$, or with an approximate decomposition $M \stackrel{\text{def}}{=}$ number of pairs of Givens rotations at refinement step
Main loop	for $k = 1, \ldots, \infty$ $\boldsymbol{y} = V^{\mathrm{H}}\boldsymbol{x}$, $\Sigma \leftarrow \sqrt{\lambda}\Sigma$
QR step and off-norm control	Use the equations in Table 1 to: • perform the QR operation • estimate the off-norm increase of all row sections • compute the number of pivots to be selected in each row section of $\Sigma(k)$ at the next (refinement) step
Refinement of Σ_S	for $m = 1{:}r$ for $n = 1{:}l_S(m)$ search the location (i,j) of the off-diagonal maximum of $\Sigma(m,1{:}r)$ choose the type of rotation as in Eq. (28) $\Sigma \leftarrow [G_{\phi_{ij}}^{i\|j}]^{\mathrm{H}}\Sigma G_{\theta_{ij}}^{i\|j}$ $V \leftarrow VG_{\theta_{ij}}^{i\|j}$ end end
Refinement of Σ_{SN}	for $m = 1{:}r$ for $n = 1{:}l_{SN}(m)$ search the location (i,j) of the maximum of $\Sigma(m,r+1{:}N)$ choose the type of rotation as in Eq. (28) $\Sigma \leftarrow [G_{\phi_{ij}}^{i\|j}]^{\mathrm{H}}\Sigma G_{\theta_{ij}}^{i\|j}$ $V \leftarrow VG_{\theta_{ij}}^{i\|j}$ end end
Refinement of Σ_N	for $m = r+1{:}N$ for $n = 1{:}l_N(m)$ search the location (i,j) of the off-diagonal maximum of $\Sigma(m,r+1{:}N)$ choose the type of rotation as in Eq. (28) $\Sigma \leftarrow [G_{\phi_{ij}}^{i\|j}]^{\mathrm{H}}\Sigma G_{\theta_{ij}}^{i\|j}$ $V \leftarrow VG_{\theta_{ij}}^{i\|j}$ end end end

distributed to the row sections of Σ_S, Σ_{SN} and Σ_N according to their specific contribution to the total norm-increase NG in $\Sigma(k)$, so that at the refinement step, more pivots are selected where a greater diagonalization effort is required. Note that depending on the off-norm distribution, it might happen that no pivot is selected in a specific row. But, in order to ensure the updating of the signal subspace projector, the number of rotations in the cross-terms region Σ_{SN} is always maintained greater or equal to 1.

Refinement: Three refinement loops are performed to reduce the off-norm of Σ_S and Σ_N, and the norm of Σ_{SN}, respectively. In each block,

a maximum search is performed in each row section only. This search is performed the number of times as prescribed at the previous step. The type of each rotation (inner or outer) is chosen as in Eq. (28) to maintain the altered singular values in decreasing ordering and maintain the signal singular values at the top positions. CSVD2 is designed to react efficiently to various tracking scenarios, as will be shown in the experiments of Section 4. If $M \cong N$ as in MA, the complexity of CSVD2 is $O(N^2)$.

3.2. *Signal subspace tracking*: *noise-averaged CSVD* (*NA-CSVD*)

NA-CSVD is the noise sphericalized version of CSVD; it tracks the signal subspace only, and an additional average noise singular $\bar{\sigma}_N$, so that the singular-value matrix is reduced to an $(r+1)\times(r+1)$ matrix $\tilde{\Sigma}(k)$. $\Sigma_N(k)$ is thus reduced to a single component $\bar{\sigma}_N$ and its diagonalization is of no concern. Also, Σ_{SN} is reduced to an $r \times 1$ column.

We present a noise sphericalized version of CSVD, the noise average-cross-terms singular-value decomposition (NA-CSVD). The NA-CSVD algorithm is presented in Table 3 and its steps are explained below.

Initialization: NA-CSVD can be initialized with an approximate sphericalized SVD, or with $V_S(0) = I_{N\times r}$ and $\tilde{\Sigma}(0) = \mathbf{0}_{r+1}$, where $I_{N\times r}$ stands for the r first columns of I_N.

Householder transformation: a Householder transformation rotates the noise singular vectors,

Table 3
NA-CSVD algorithm

Step	Operation
Initialization	Initialize with $V_S = I_{N\times r} \quad \tilde{\Sigma} = \mathbf{0}_{r+1}$ or with an approximate sphericalized SVD
Loop	for $k = 1, \ldots, \infty$ $\bar{\sigma}_N = \tilde{\Sigma}(r+1, r+1)$
Householder transformation	$\boldsymbol{x}_S = V_S^{\mathrm{H}}\boldsymbol{x}$ $\boldsymbol{z} = \boldsymbol{x} - V_S\boldsymbol{x}_S$ $\boldsymbol{v}_N = \boldsymbol{z}/\|\boldsymbol{z}\|$
QR step	$\boldsymbol{y}^{\mathrm{H}} \leftarrow [\boldsymbol{x}_S^{\mathrm{H}}, \|\boldsymbol{z}\|], \quad \tilde{\Sigma} \leftarrow \sqrt{\lambda}\tilde{\Sigma}$ for $i = 1{:}r+2$ compute the angle ϕ_i to zero $\boldsymbol{y}(i)$ as in $[G_{\phi_i}^{1\|2}]^{\mathrm{H}}\begin{bmatrix}\sigma_{ii}\\ \boldsymbol{y}(i)\end{bmatrix} = \begin{bmatrix}\sigma'_{ii}\\ 0\end{bmatrix}$ $\begin{bmatrix}\tilde{\Sigma}\\ \boldsymbol{y}^{\mathrm{H}}\end{bmatrix} \leftarrow [G_{\phi_i}^{i\|r+1}]^{\mathrm{H}}\begin{bmatrix}\tilde{\Sigma}\\ \boldsymbol{y}^{\mathrm{H}}\end{bmatrix}$ end
Refinement step	for $l = r$ downto 1 choose the type of rotation as in Eq. (28) $\tilde{\Sigma} \leftarrow [G_{\phi_i}^{l\|r+1}]^{\mathrm{T}}\tilde{\Sigma}G_{\theta_i}^{l\|r+1}$ $[V_S, \boldsymbol{v}_N] \leftarrow [V_S, \boldsymbol{v}_N]G_{\theta_i}^{l\|r+1}$ end
Sphericalization	$\tilde{\Sigma}(r+1, r+1) \leftarrow \sqrt{\dfrac{(N-r-1)\lambda\bar{\sigma}_N^2 + \tilde{\Sigma}(r+1, r+1)^2}{N-r}}$ end

so that the projection of the measurement vector $\boldsymbol{x}(k)$ onto the noise subspace is parallel to the first noise vector, $\boldsymbol{v}_N$ (see [4,11] for details).

QR step: the QR step is achieved as in Eq. (4) but with only $r+1$ Givens rotations.

Refinement: the r pivots are chosen from the rth entry up to the first entry of the $(r+1)$th column of $\tilde{\Sigma}(k)$ to provide a highest EGR of 1. For each pair of rotations, the type of rotations is chosen as in Eq. (28) to maintain the decreasing ordering of the altered singular values.

Sphericalization: finally, the average noise singular value is re-estimated.

Here, the full cross-terms cancellation concept leads to a type of RO-FST structure. However, our approach is based on a different type of decomposition, namely UXV, and both algorithms have different refinement structures. The complexity of NA-CSVD is O(Nr).

3.3. Further remarks on the proposed algorithms

Regarding the numerical properties of the algorithms, let us note that CSVD2 and NA-CSVD retain the classical QR + refinement structure from [17] and [12], but simply apply the rotations in a different way. Therefore, an error propagation analysis similar to the one presented in [17] can be formulated; the algorithms can be proven to be numerically stable, provided the forgetting factor λ verifies $\lambda < 1$, and if a reorthogonalization procedure is included.

Our innovative way of monitoring the off-norm increase throughout $\Sigma(k)$ allows us to dynamically

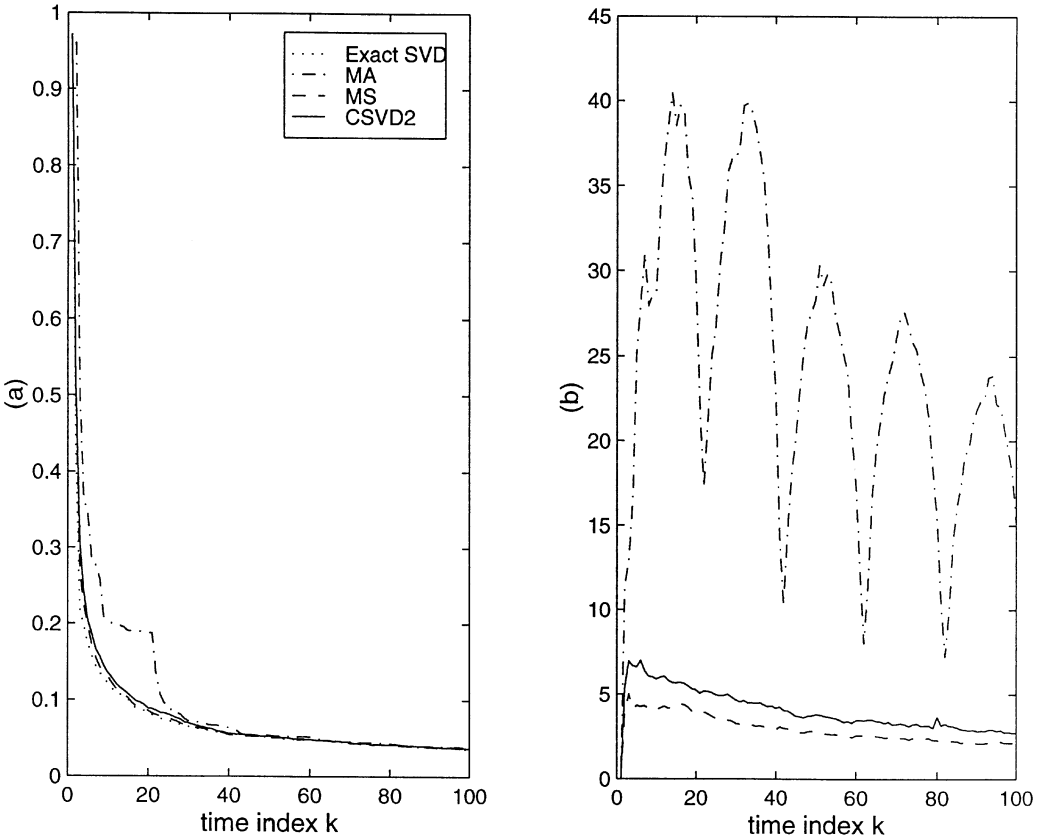

Fig. 2. Initial convergence of complete tracking algorithms: Exact SVD, CSVD2, MA and Maximum search (MS): (a) tracking error, (b) off-norm of $\Sigma(k)$. $N=20$, $r=2$, $\theta_1=10^\circ$, $\theta_2=20^\circ$, SNR = 10 dB, $\lambda=0.99$, 10 experiments average.

distribute the Givens rotations between each row in Σ_S, Σ_N and Σ_{SN} so that the off-norm is reduced where it is predicted to grow more significantly. The main result is that CSVD2 can achieve a tracking performance similar to Moonen's algorithm (MA) [17] for instance, but with much fewer Givens rotations at the refinement step, as shown in the simulation experiments.

Unlike the rank revealing URV updating algorithm [30], CSVD2 does not require the initial conditions regarding the size of the entries of each block to be explicitely stated. The block partitioning of $\Sigma(k)$, or similarly the sorting of V into $[V_S V_N]$ is obtained after a few iterations and maintained by choosing the appropriate type of each rotation. Therefore, CSVD2 maintains the signal singular values at the top position and confines Σ_{SN} in the upper-right corner of $\Sigma(k)$, as in Stewart's URV.

Regarding the issue of sudden rank changes, one may use the MDL criteria [33] jointly with CSVD2 to estimate the rank variations. NA-CSVD has been succesfully coupled with a recent rank tracking technique for spherical subspace trackers (RSST) proposed by Kavcic and Yang [13]. NA-CSVD/RSST tracks efficiently the subspaces and the variations of the rank changes in various tracking scenarios, including moving and crossing sources [21].

4. Experimental results

In this section, computer simulations are conducted to test the new algorithms and the underlying theory regarding the efficient use of Givens rotations in QR Jacobi-type subspace tracking algorithms. In all the experiments, we deal with the problem of estimating the direction of arrival (DOAs) of r incident plane waves on a receiving linear array of N sensors. The intersensor spacing

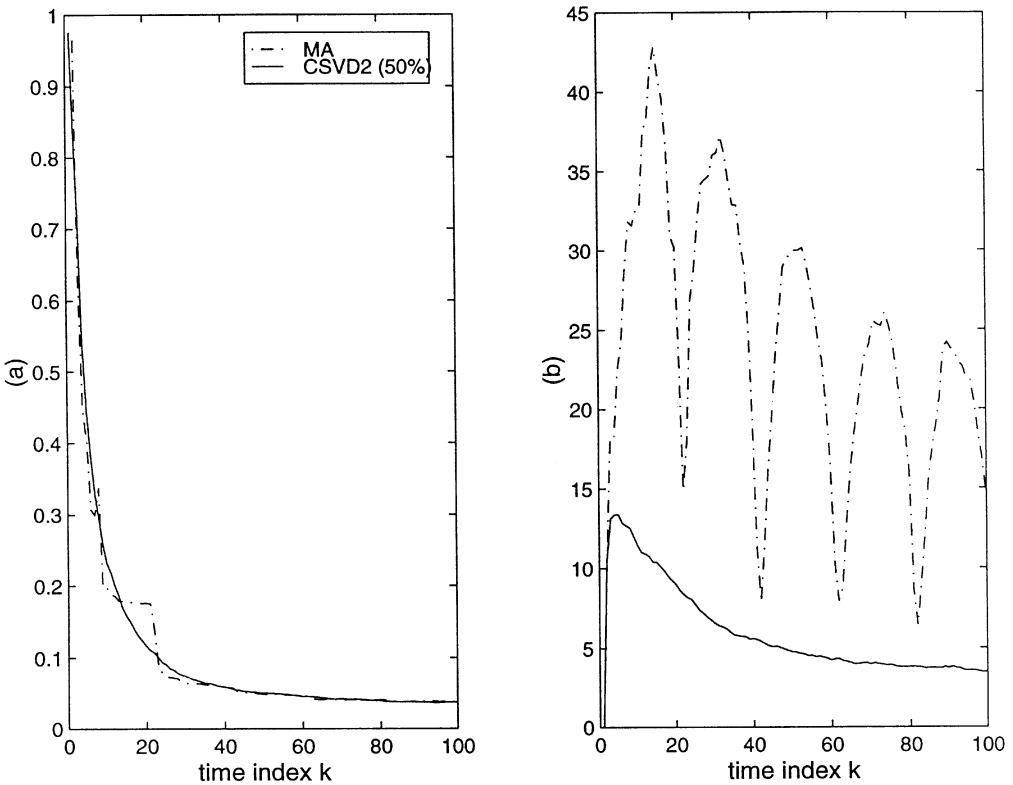

Fig. 3. Comparison of MA ($M = N - 1 = 19$), and CSVD2 using only 50% ($M = 9$) of the refinement rotations used in MA: (a) tracking error, (b) off-norm of $\Sigma(k)$. $N = 20$, $r = 2$, $a_1 = 10°$, $a_2 = 20°$, SNR $= 10$ dB, $\lambda = 0.99$, 10 experiments average.

d is set to half the wavelength. At each snapshot, the measurement vector $\boldsymbol{x}(k)$ is obtained and its ith component represents the combined action of all the sources at the ith sensor, as in

$$x_i(k) = \sum_{l=1}^{r} s_l(k)\mathrm{e}^{\mathrm{j}(i-1)\omega_l(k)} + n_i(k), \quad i = 1, \ldots, N, \tag{31}$$

where $\omega_l(k)$ is the electrical angle (intersensor phase-shift at time index k); $s_l(k)$ is the complex amplitude of the lth source; $n_i(k)$ is an additive background noise component. $n_i(k)$ and $s_l(k)$ are modelled as independent zero-mean complex circular Gaussian variables. The variance of $n_i(k)$ is set to 1, and the complex amplitudes $s_l(k)$ correspond to each of the sources signal-to-noise ratio (SNR).

Our goal is to show how the performance of existing QR Jacobi-type subspace tracking algorithms can be reached by using efficiently fewer pairs of Givens rotations at the refinement step. We ran two series of experiments, one for each type of QR-Jacobi-type algorithms: *complete tracking* algorithms which track both the signal and noise subspaces, and *partial tracking* algorithms which track the signal subspace only by sphericalizing the noise subspace. All algorithms are initialized with an initial exact decomposition obtained from a few initial measurement samples. We would like however to mention that CSVD2 and NA-CSVD provides similar experimental results if initialized with $V(0) = I$ and $\Sigma(0) = \mathbf{0}$.

Experiment 1. *Initial convergence of complete tracking algorithms.* $N = 20$ sensors are used to track $r = 2$ stationary sources at $a_1 = 10^\circ$ and $a_2 = 20^\circ$ with SNR $= 10\,\mathrm{dB}$. The forgetting factor λ is set to 0.99. The experiment involves the following

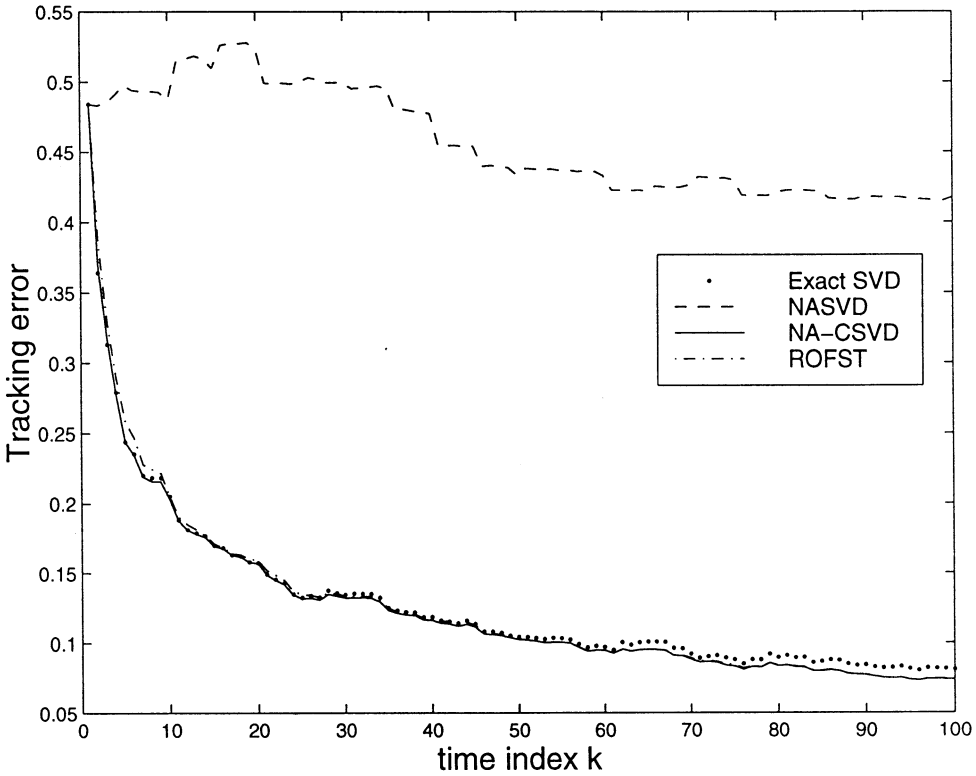

Fig. 4. Initial convergence of partial tracking algorithms: NA-CSVD, RO-FST (one refinement), Exact SVD and NASVD. $N = 10$, $r = 5$, SNR $= 10\,\mathrm{dB}$, $a_1 = 10^\circ$, $a_2 = 20^\circ$, $a_3 = 30^\circ$, $a_4 = 40^\circ$, $a_5 = 50^\circ$, $\lambda = 0.99$, 10 experiments average.

algorithms: Exact SVD which computes the exact SVD of Eq. (1) at each time iteration, Moonen's algorithm (MA) where the refinement consists in $M = N - 1 = 19$ pairs of rotations, Maximum Search (MS) which searches the maximum of $\Sigma(k)$ prior to each of the $M = N - 1 = 19$ refinement rotations (as detailed in Section 2.2.1) and CSVD2 ($M = N - 1 = 19$). All algorithms are initialized with the exact SVD obtained from only one initial sample $\boldsymbol{x}(0)$, i.e. $A(0) = \boldsymbol{x}(0)^{\mathrm{H}} = U(0)_{1\times 1}\Sigma(0)_{1\times N} V(0)^{\mathrm{H}}_{N\times N}$ with $\Sigma(0) = [\|\boldsymbol{x}(0)\|, 0 \cdots 0]$; (to start with a square singular value matrix, we use $\Sigma(0) \leftarrow \mathrm{diag}(\Sigma(0))$ instead. Fig. 2(a) and (b) show the initial convergence for the tracking error and the off-norm of $\Sigma(k)$ as defined in Eqs. (9) and (22), respectively; these results are the average of a 10-trial experiment.

Regarding MA, one notices an abrupt decrease of the tracking error of MA is observed each 20 samples only, i.e. at $k = 2$ and $k = 22$. This is due to the permutation properties of outer rotations; every N time iterations, the signal singular vectors (and values) are properly positioned and for the r following time iterations, 18 of the 19 pairs of rotations in the refinement step annihilate a cross-term. The same effect is more visible in the off-norm of the singular value matrix from MA. Since MA requires N measurement samples to zero all upper-diagonal entries, the off-norm of $\Sigma(k)$ is really lowered each N samples only, i.e. at specific time indexes $k = 2$, $k = 22$, $k = 42, \ldots$ and so on. The increase of the off-norm between these time indexes is due to the fact that the QR step constantly contributes to the increase of upper-diagonal entries which have to wait N time iterations before being zeroed.

The MS algorithm maintains the off-norm of $\Sigma(k)$ at the lowest level, but within a complexity of $O(N^3)$. CSVD2 achieves almost the same diagonalization performance as MS by targetting the annihilation of large off-diagonal entries, but with a smaller searching complexity cost. This confirms that our estimation of the off-norm distribution in

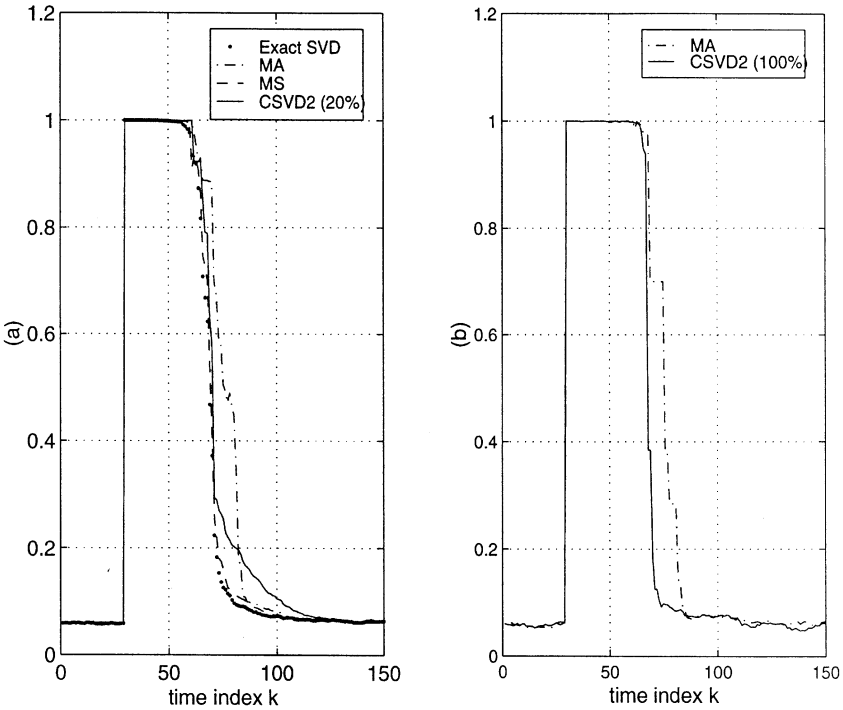

Fig. 5. Tracking error of complete tracking algorithms: sudden subspace rotation. Exact SVD, CSVD2, MS and MA. $N = 20$, $r = 5$, SNR = 10 dB, $\lambda = 0.96$, average of 10 experiments: (a) CSVD2 uses only 20% of $N - 1$ Givens rotations ($M = 9$), (b) CSVD2 uses 100% of the $N - 1$ Givens rotations ($M = 19$).

$\Sigma(k)$ is precise. Note that, even with a loss of information at the QR step, CSVD2 reaches the subspace tracking performance of an exact SVD. In Fig. 3(a) and (b), we compare MA ($M = 19$) with CSVD2 using only $M = 9$ pairs of rotations at the refinement step (approximatively 50% of the refinement 19 pairs of rotations in MA). It shows that even with fewer Givens rotations, CSVD2 tracks the subspaces and diagonalizes $\Sigma(k)$ more efficiently than MA.

Experiment 2. *Initial convergence of partial tracking algorithms*. Along with the exact SVD, we simulate various partial tracking algorithms: NASVD (the noise sphericalized version of MA), RO-FST (one refinement only) and proposed NA-CSVD. We limited this experiment to those three algorithms since [22] offers numerous comparison tests between RO-FST and other well-known algorithms, including ROSA [4], Karasalo's algorithm [11] and TQR-SVD [6]. The simulation parameters are: $N = 10$, $r = 5$, SNR $= 10$ dB, $a_1 = 10^\circ$, $a_2 = 20^\circ$, $a_3 = 30^\circ$, $a_4 = 40^\circ$, $a_5 = 50^\circ$, $\lambda = 0.99$. Fig. 4 shows the tracking error during initial convergence, average of 10 experiments. There is almost no difference between NA-CSVD, RO-FST and the exact SVD. This confirms that the uppper-triangular structure is not required, since the main difference between RO-FST and NA-CSVD is the structure of $\Sigma(k)$ and consequently the refinement step. As expected, NASVD suffers from the initialization starting-point which has been obtained with only six samples. Also, since its tracking error is modified by only one pair of rotations at the refinement step, its convergence is slow, compared to the other cross-terms based algorithms, i.e. NA-CSVD and RO-FST.

Experiment 3. *Complete tracking algorithms and sudden subspace rotation*. $N = 20$, $r = 5$, SNR $= 10$ dB,

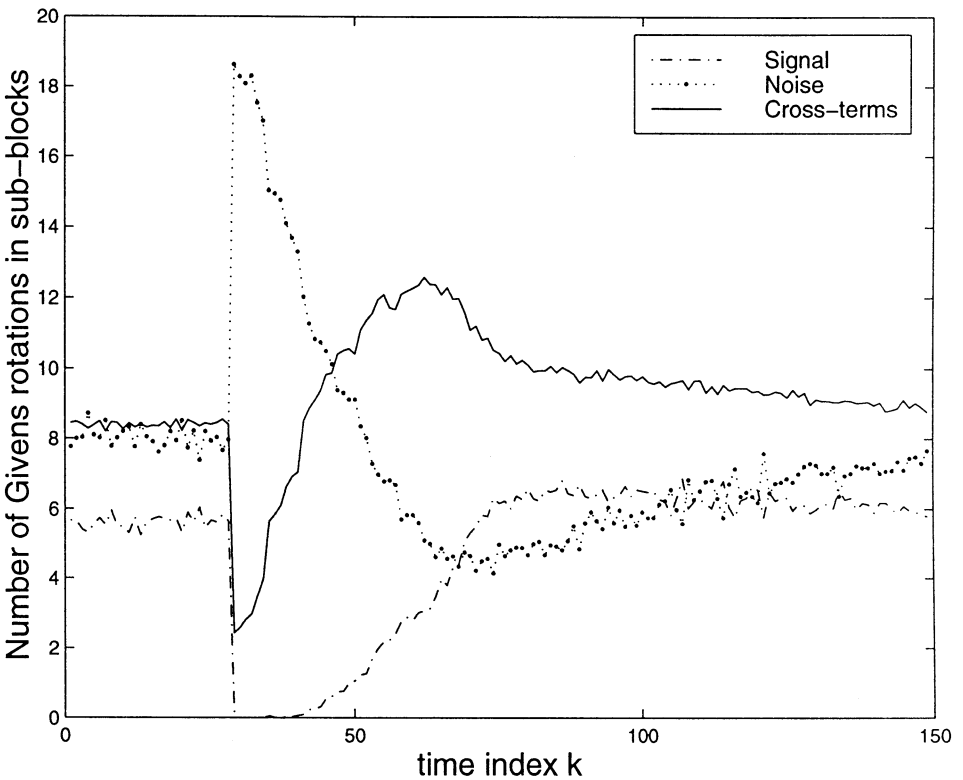

Fig. 6. Number of Givens rotations applied by CSVD2 (100%) in $\Sigma_S(k)$ (Signal), $\Sigma_N(k)$ (Noise) and Σ_{SN} (Cross-terms) averaged over 40 trials. The simulation parameters are as in Fig. 5.

$\lambda = 0.96$. The algorithms are initialized with an exact SVD obtained from the following DOAs: $a_1 = 10^\circ$, $a_2 = 20^\circ$, $a_3 = 30^\circ$, $a_4 = 40^\circ$ and $a_5 = 50^\circ$. At $k = 30$, we remove these sources and introduce new ones with opposite DOAs $a_1 = -10^\circ$, $a_2 = -20^\circ$, $a_3 = -30^\circ$, $a_4 = -40^\circ$ and $a_5 = -50^\circ$. This causes the signal subspace associated with the incoming data to rotate by 90°. We then observe the length of time each algorithm takes to stabilize itself, i.e. to rotate the approximate subspaces in the orthogonal direction. The tracking error of the algorithms are in Fig. 5(a), where the refinement step of CSVD2 consists in only $M = 4$ pairs of rotations (approximatively 20% of the usual 19 pairs of rotations) in its refinement step. CSVD2 appears as a robust algorithm, even when it uses fewer pairs of rotations at the refinement step. Fig 5(b) shows the tracking error of CSVD2 compared to MA, with the same number of pairs of Givens rotations at the refinement step, i.e. $M = 19$. Fig. 6 shows the number of pairs of rotations applied by CSVD2 ($M \cong 19$) to the *Signal* block $\Sigma_S(k)$, the *Noise* block $\Sigma_N(k)$ and the *Cross-terms* block (Σ_{SN}), averaged over 40 trials. At $k = 30$, the proportion of rotations applied to Σ_N increases to respond to the off-norm increase in $\Sigma_N(k)$. This experiment shows that CSVD2 effectively applies the rotations where a diagonalization procedure is required, without any prior information of the DOA changes. The direct consequence is the robustness of CSVD2 which stabilizes itself faster.

Experiment 4. *Partial tracking algorithms and sudden subspace rotation.* The simulation parameters are the same as in Experiment 3. Fig. 7 shows the tracking error. We note once again the superiority of cross-terms based algorithms (NA-CSVD and RO-FST) over NASVD. Note that under the same experiment conditions, spherical subspace trackers

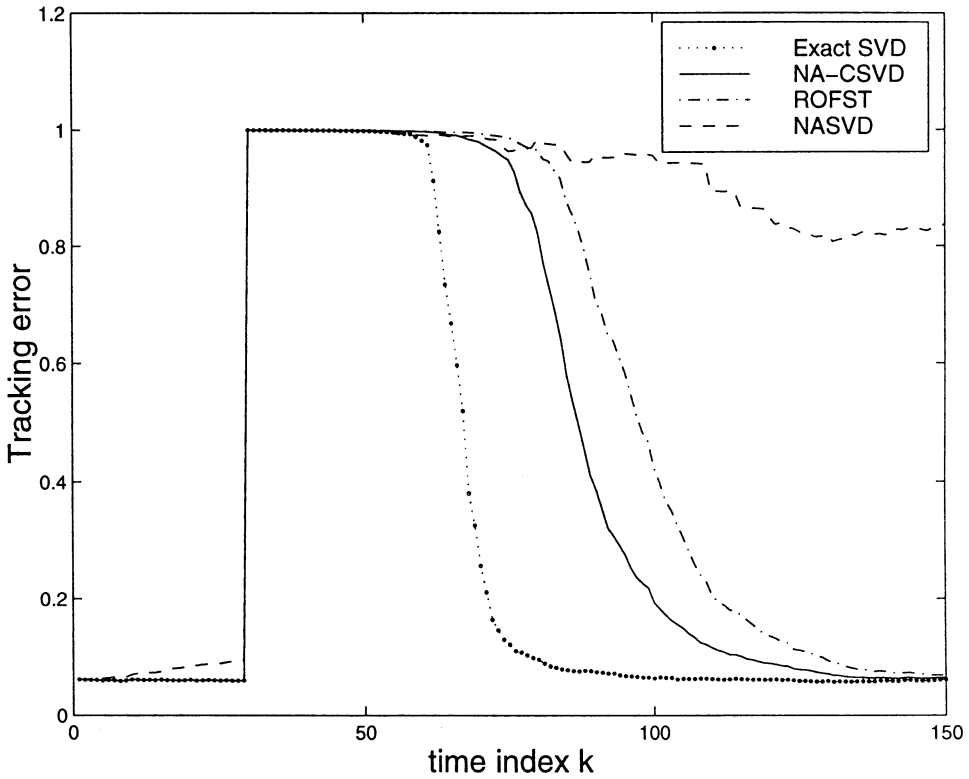

Fig. 7. Tracking error of partial tracking algorithms: sudden subspace rotation. $N = 20$, $r = 5$, SNR = 10 dB, $\lambda = 0.96$, 10 experiments average.

face drastic DOA changes with less robustness if compared to complete tracking algorithms; they require a longer period to stabilize themselves.

Experiment 5. *Complete tracking algorithms and non-stationary data.* $N = 40$ sensors are used to track $r = 2$ sources, $a_1 = (10 + 0.04k)^\circ$ and $a_2 = (30 - 0.04k)^\circ$ which cross at $k = 250$ with SNR $= 10$ dB and $\lambda = 0.92$. All the algorithms are initialized with the exact SVD computed with 20 previous samples. For each algorithm, roots-MUSIC [1] is used to estimate the DOAs at time index k. Fig. 8 shows the tracked angles averaged over 10 trials. One can clearly see how the choice of the rotations type coupled with the choice of the pivots location can greatly influence the parameters tracking performance. MA is influenced by the permutations induced by outer rotations. Indeed, in each block of 40 consecutive samples, the tracking is effective during the first two samples, and almost stall during the 38 following ones; the result is a stairs-like angle estimate. What may be considered as an advantage in Experiment 1 (initial convergence) is a drawback when the experiment involves non-stationary data. On the other hand, CSVD2 tracks continuously the sources DOAs, even with fewer Givens rotations at the refinement step, i.e. only 20% ($M = 8$) of the 39 pairs of Givens rotations used in MA.

Experiment 6. *Partial tracking algorithms and non-stationary data.* Here, we have $N = 8$, $r = 2$, $\lambda = 0.95$, SNR $= 5$ dB, $a_1 = (20 + 0.01k)^\circ$ and $a_2 = (30 - 0.01k)^\circ$ cross at $k = 500$. Fig. 9 shows the estimated DOAs from one realization only. As expected, the cross-terms based algorithms, i.e. NA-CSVD and RO-FST, perform better and provide more accurate DOA estimates. Fig. 10 shows the tracking error averaged over 20 experiments; it indicates that the algorithms face a stronger perturbation when the sources get closer. Here, the superior performance of cross-terms based

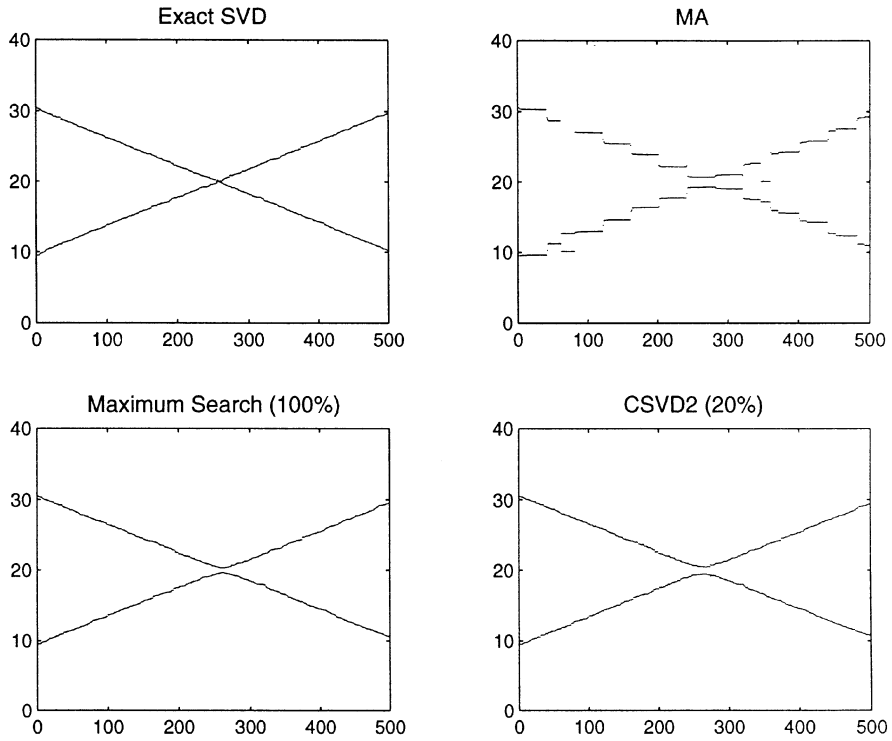

Fig. 8. Angles estimates by complete tracking algorithms: crossing sources. Exact SVD, Maximum Search (MS), CSVD2 (20%) and MA. $N = 20$, $r = 2$, $a_1 = (10 + 0.04k)^\circ$, $a_2 = (30 - 0.04k)^\circ$, SNR $= 10$ dB, $\lambda = 0.92$, 10 experiments average.

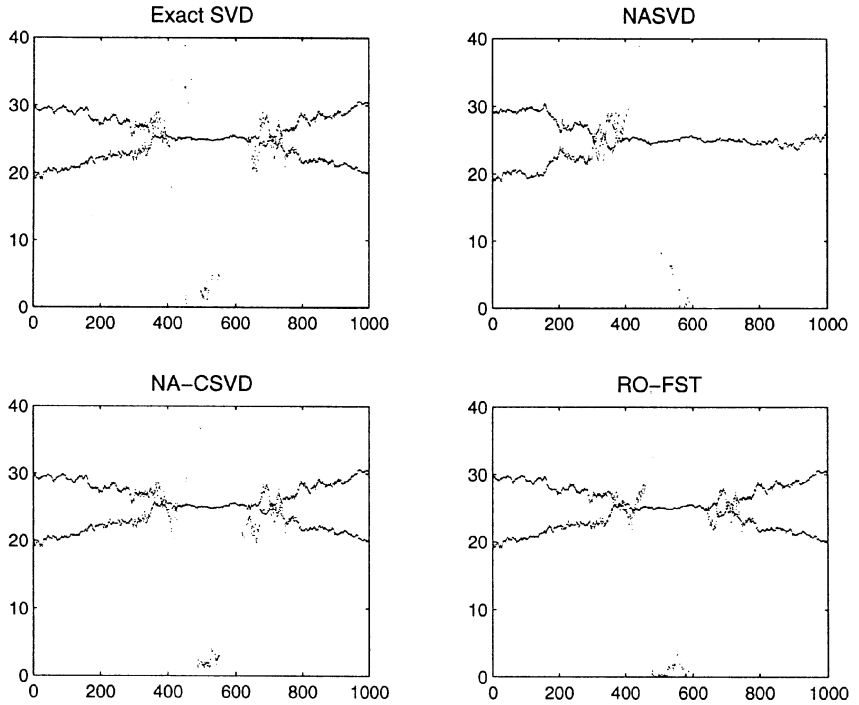

Fig. 9. Angles estimates by sphericalized noise subspace algorithms: crossing sources. $N = 8$, $r = 2$ crossing sources, $\lambda = 0.95$, SNR $= 10$ dB, $a_1 = (20 + 0.01k)^\circ$ and $a_2 = (30 - 0.01k)^\circ$ cross at $k = 500$, 1 experiment.

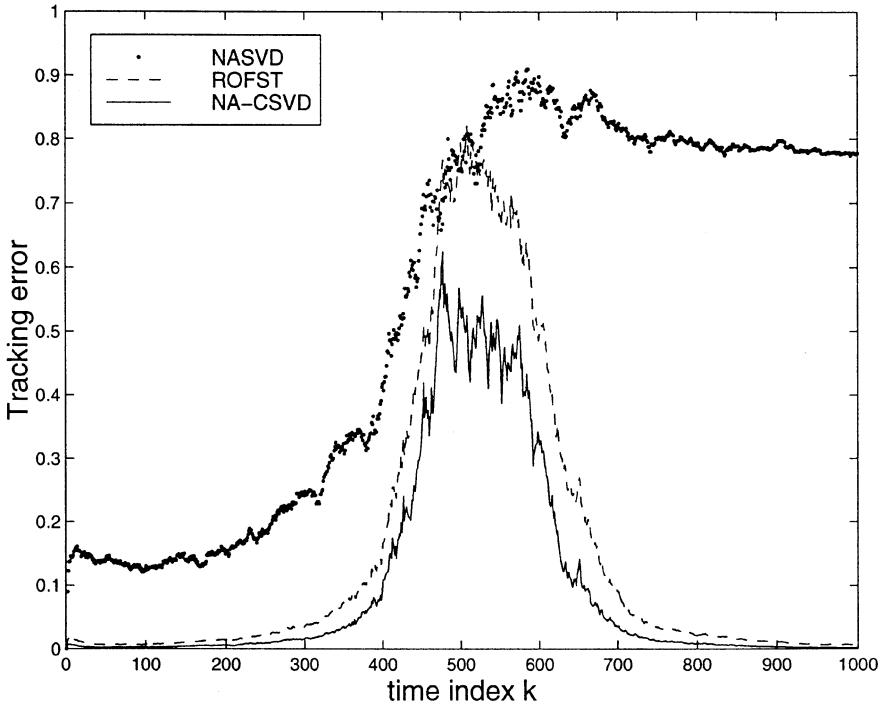

Fig. 10. Tracking error of partial tracking algorithms, averaged over 20 experiments: crossing sources. Exact SVD, NASVD, RO-FST and NA-CSVD; $N = 8$, $r = 2$ crossing sources $a_1 = (20 + 0.01k)^\circ$ and $a_2 = (30 - 0.01k)^\circ$ cross at $k = 500$, $\lambda = 0.93$, SNR $= 10$ dB.

algorithms confirm the need to locate pivots almost exclusively in Σ_{SN}. More particularly, NA-CSVD outperforms RO-FST and NASVD.

5. Conclusion

In this paper, we investigated the optimal use of Givens rotation as both a diagonalization tool and a vector rotation tool. On one hand we showed how a better diagonalization of the singular-value matrix is obtained when the Givens rotations are applied so as to annihilate the largest entries of the singular-value matrix. To reduce the complexity of this approach, we proposed an efficient method which controls the off-norm in $\Sigma(k)$ by distributing dynamically the Givens rotations. On the other hand, we demonstrated the need to zero cross-terms entries in the case of subspace tracking. We showed how the choice for inner or outer rotations can be made to maintain the ordering of the singular values. Finally, based on this study, we proposed two efficient and robust subspace tracking algorithms, CSVD2 and NA-CSVD. As shown in simulation experiments, the proposed algorithms outperform existing QR Jacobi-type algorithms, even with fewer Givens rotations.

Acknowledgements

This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. Philippe Pango acknowledges a fellowship from the Government of Côte d'Ivoire. The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] A. Barabell, Improving the resolution performance of eigenstructure-based direction-finding algorithms, in: Proc. Internat. Conf. Acoustic Speech and Signal Processing, 1983, pp. 336–339.

[2] C.H. Bischof, G.M. Shroff, On updating signal subspaces, IEEE Trans. Signal Process. 40 (1) (1992) 96–105.

[3] B. Champagne, Adaptive eigendecomposition of data covariance matrices based on first-order perturbations, IEEE Trans. Signal Process. 42 (10) (October 1994) 2758–2770.

[4] R.D. Degroat, Noniterative subspace tracking, IEEE Trans. Signal Process. 40 (3) (March 1992) 571–577.

[5] R.D. DeGroat, E.M. Dowling, H. Ye, D.A. Linebarger, Spherical subspace tracking for efficient, high performance adaptive signal processing applications, Signal Processing 50 (1996) 101–121.

[6] E.M. Dowling, L.P. Ammann, R.D. DeGroat, An adaptive TQR-SVD for angle and frequency tracking, IEEE Trans. Signal Process. 42 (4) (April 1994) 914–926.

[7] R.D. Fierro, Perturbation analysis for two-sided (or complete) orthogonal decompositions, SIAM J. Matrix Anal. Appl. 17 (2) (April 1996) 383–400.

[8] R.D. Fierro, J.R. Bunch, Bounding the subspaces from rank-revealing two-sided orthogonal decompositions, SIAM J. Matrix Anal. Appl. 16 (3) (July 1995) 743–759.

[9] G.H. Golub, C.F. Van Loan, Matrix Computations, 2nd ed., John Hopkins Univ. Press, Baltimore, MD, 1989, pp. 444–445.

[10] S. Haykin, Adaptive Filter Theory, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991, p. 426.

[11] I. Karasalo, Estimating the covariance matrix by signal subspace averaging, IEEE Trans. Accoust. Speech Signal Process. 34 (February 1986) 8–12.

[12] A. Kavcic, B. Yang, A new efficient subspace tracking based on singular value decomposition, in: Proc. Internat. Conf. on Acoustic Speech and Signal Processing, Vol. IV, 1994, pp. 485–488.

[13] A. Kavcic, B. Yang, Adaptive rank estimation for spherical subspace trackers, IEEE Trans. Signal Process. 44 (6) (June 1996) 206–217.

[14] K. Konstantinides, K. Yao, Statistical analysis of effective singular values in matrix rank determination, IEEE Trans. Accoust. Speech Signal Process. 36 (5) (May 1988) 757–763.

[15] R. Kumaresan, D.W. Tuffs, Estimating the angles of arrival of multiple plane waves, IEEE Trans. Aerospace Electron. Systems AES-19 (January 1982) 134–139.

[16] F.T. Luk, A triangular processor array for computing the singular values, Linear Algebra Appl. 77 (1986) 259–273.

[17] M. Moonen, Van Dooren, J. Vandewalle, A singular value decomposition algorithm for subspace tracking, SIAM Matrix Anal. Appl. 13 (4) (October 1992) 1015–1038.

[18] N.L. Owsley, Adaptive data orthogonalization, in: IEEE Internat. Conf. on Acoustic Speech and Signal Processing, Tulsa, 1978, pp. 109–112.

[19] P.A. Pango, Algorithmes rapides de mise à jour de décompositions matricielles et application au traitement de signaux multi-dimensionnels, Ph.D Dissertation (to be published), INRS Telecommunications, 1998.

[20] P.A. Pango, B. Champagne, Accurate subspace tracking algorithms based on cross-space properties, in: Proc. Internat. Conf. on Acoustic Speech and Signal Processing, Munich, Vol. 5, 1997, pp. 3833–3836.

[21] P.A. Pango, B. Champagne, On the issue of rank estimation in subspace tracking: the NA-CSVD solution, in: Proc. European Signal Processing Conf. EUSIPCO-98, Rhodes, Vol. 3, 1998, pp. 1773–1776.
[22] D.J. Rabideau, Fast, rank adaptive subspace tracking, IEEE Trans. Signal Process. 44 (9) (September 1996) 2229–2244.
[23] D.J. Rabideau, A.O. Steinhardt, Fast subspace tracking, in: Proc. 7th Signal Processing, Workshop on SSAP, June 1994.
[24] R. Roy, T. Kailath, Estimation of signal parameters via rotational invariance techniques, IEEE Trans. Accoust. Speech Signal Process. 37 (7) (July 1989) 984–995.
[25] R.O. Schmidt, Multiple emitter location and signal parameter estimation, in: Proc. RADC Spectrum Estimation Workshop, Rome, New York, 1979, pp. 243–258; reprinted in IEEE Trans. Antennas and Propagation 34 (3) (March 1986) 276–280.
[26] R. Schreiber, Implementation of adaptive array algorithms, IEEE Trans. Accoust. Speech Signal Process. 34 (October 1986) 1038–1045.
[27] Shen-Fu Hsiao, Adaptive Jacobi method for parallel singular value decompositions, in: Proc. Internat. Conf. on Acoustic Speech and Signal Processing, Vol. V, 1995, pp. 3203–3206.
[28] G.W. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, SIAM Rev. 15 (1973) 727–764.
[29] G.W. Stewart, A Jacobi-like algorithm for computing the Schur decomposition of a Non-Hermitian matrix, SIAM J. Sci. Stat. Comput. 6 (4) (October 1985) 853–864.
[30] G.W. Stewart, An updating algorithm for subspace tracking, IEEE Trans. Signal Process. 40 (6) (June 1992) 1535–1541.
[31] G.W. Stewart, Updating a rank-revealing ULV decomposition, SIAM J. Matrix Anal. Appl. 4 (1993) 494–499.
[32] J.E. Volder, The CORDIC trigonometric computing technique, IRE Trans. Electron. Comput. 8 (3) (September 1959) 330–334.
[33] M. Wax, T. Kailath, Detection of signals by information theoretic criteria, IEEE Trans. Signal Process. 33 (1985) 387–392.
[34] B. Yang, Projection approximation subspace tracking, IEEE Trans. Signal Process. 43 (1) (January 1995) 95–107.
[35] J. Yang, M. Kaveh, Adaptive eigensubspace algorithms for direction or frequency estimation and tracking, IEEE Trans. Accoust. Speech Signal Process. 36 (2) (February 1988) 241–251.