A fast algorithm for subspace tracking

Nicola Mastronardi
Marc Van Barel
Raf Vandebril

Report TW430, May 2005

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A fast algorithm for subspace tracking

Nicola Mastronardi
Marc Van Barel
Raf Vandebril

Report TW430, May 2005

Department of Computer Science, K.U.Leuven

Abstract

Fast estimation and tracking of the principal subspace of a sequence of random vectors is a classic problem, widely encountered in areas such as radar, sonar and speech processing, data compression, data filtering, parameter estimation, pattern recognition, neural analysis, wireless communications, to name just a few.

Among the most robust algorithms for subspace tracking there are the so called OPERA–based algorithms with computational complexity $2nr^2 + O(r^2)$, where n is the input vector dimension and r $(n \gg r)$ is the desired number of eigencomponents.

In this paper we propose a fast and stable algorithm for subspace tracking based on the EVD–OPERA algorithm with $6nr+15r^2$ computational complexity.

Keywords : subspace tracking, rank–one updating, eigenvalue decompositon.
AMS(MOS) Classification : Primary : 15A18, Secondary : 65Y20.

A Fast Algorithm for Subspace Tracking

Nicola Mastronardi, Marc Van Barel, *Member, IEEE*, Raf Vandebril

Abstract

Fast estimation and tracking of the principal subspace of a sequence of random vectors is a classic problem, widely encountered in areas such as radar, sonar and speech processing, data compression, data filtering, parameter estimation, pattern recognition, neural analysis, wireless communications, to name just a few.

Among the most robust algorithms for subspace tracking there are the so called OPERA–based algorithms with computational complexity $2nr^2 + O(r^2)$, where n is the input vector dimension and r $(n \gg r)$ is the desired number of eigencomponents.

In this paper we propose a fast and stable algorithm for subspace tracking based on the EVD–OPERA algorithm with $6nr + 15r^2$ computational complexity.

Index Terms

subspace tracking, rank–one updating, eigenvalue decompositon.

I. Introduction

FAST estimation and tracking of the principal subspace of a sequence of random vectors is a classic problem, widely encountered in areas such as radar, sonar and speech processing, data compression, data filtering, parameter estimation, pattern recognition, neural analysis, wireless communications (see, e.g. [1], [3], [4], [8], [12], [13], [18]–[20], [24] and the references therein).

Among the most robust algorithms for subspace tracking there are the so called OPERA–based algorithms [10], [12], [13], [20]. Indeed, OPERA–based algorithms have been shown to be "locally" more robust than, e.g., PAST–based algorithms [24], provided the initial basis of the principal subspace is "close" to the exact one [9].

There are mainly two equivalent implementations of the OPERA–based algorithms, one based on the singular value decomposition of the signal subspace formed from the observations of a non stationary signal in non stationary noise, called SVD–OPERA, and one based on the eigenvalue decomposition of the covariance matrix formed from the observations of a non stationary signal in non stationary noise, called EVD–OPERA. Both have the same computational complexity $2nr^2 + O(r^2)$, where n is the input vector dimension and r $(n \gg r)$ is the desired number of eigencomponents[1].

In this paper we propose a fast algorithm for tracking the signal subspace of a sequence of vectors. At each iteration of the new method, the smallest eigenvalue and the corresponding eigenvector of a Hermitian tridiagonal matrix modified by a Hermitian rank–one matrix, is computed. Then a unitary matrix, approximation of the original signal subspace, is multiplied by a Householder matrix. Each iteration requires $6nr + 15r^2$ floating point operations.

Moreover, we will show that one step of the new EVD–OPERA is equivalent to one step of the EVD–OPERA algorithm.

For the sake of brevity we describe only the algorithm based on EVD–OPERA. In a similar way, a fast algorithm can be derived for SVD-OPERA based algorithms.

The paper is organized as follows. In the next section the general framework of subspace tracking algorithms and the EVD–OPERA algorithm are described. The new fast algorithm for tracking an othogonal subspace of a sequence of vectors is described in section III. The comparison between the performances of the EVD–OPERA algorithm and the new algorithm can be found in section IV followed by the conclusions.

II. Subspace tracking procedure

Let $r \ll n$ be the dimension of the signal subspace and consider a set of $k \geq r$ distinct data snapshots $x_i \in \mathbb{C}^n$, $i = 1, \ldots, k$. For instance, in direction of arrival (DOA) estimation [12], r represents the number of sources present. The rank of the data snapshots $X = [x_1, x_2, \ldots, x_k]$ is equal to r if noise is absent. Unfortunately, the matrix X has generally full rank because of

Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, via G. Amendola 122/D, I–70126 Bari, Italy (Email: n.mastronardi@area.ba.cnr.it).

The research of this author was partially supported by MIUR, grant number 2004015437, by the short term mobility program, Consiglio Nazionale delle Ricerche and by VII Programma Esecutivo di Collaborazione Scientifica Italia–Comunità Francese del Belgio, 2005–2006.

Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Email: {Marc.VanBarel,Raf.Vandebril}@cs.kuleuven.ac.be)

The research of the second and third author was supported by G.0176.02 (Asymptotic analysis of the convergence behavior of iterative methods in numerical linear algebra), and G.0184.02 (CORFU: Constructive study of Orthogonal Rational Functions), by the K.U.Leuven (Bijzonder Onderzoeksfonds), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling).

[1]In this paper, to compute the computational complexity and the storage, we indicate only the term of higher order, neglecting those of lower order, e.g., if the complexity is $4n^2 + 3n$, we say that the complexity is $4n^2$.

the noise. The vector space S_X, representing the column space of X, is commonly referred as the *signal subspace* and its orthogonal complement $S_X^\perp$ the *noise subspace*. When noise is present, only approximations to S_X and $S_X^\perp$ can be computed from the measured data. One method to estimate the rank of the matrix X [11], [12], requires an estimate for the covariance matrix, given by the matrix

$$\hat{C} = \frac{1}{k}\sum_{i=1}^{k} x_i x_i^H = \frac{1}{k} X X^H \tag{1}$$

Since the estimated covariance matrix $\hat{C}$ in (1) is a Hermitian positive semidefinite matrix, it has non–negative real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 0$ with corresponding eigenvectors $v_1, v_2, \ldots, v_k$,

$$\hat{C} = V \Lambda V^H = \sum_{i=1}^{k} \lambda_i v_i v_i^H$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$ and $V = [v_1, v_2, \ldots, v_k]$. Hence, if there is no noise the rank of $\hat{C}$ is exactly r, i.e., $\lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_k = 0$. Due to the noise, the matrix $\hat{C}$ has rank k and r must be estimated in some way. Statistically based methods for such estimations can be found in [11]. Once the value of r has been estimated, the estimate of the signal subspace can be obtained from the matrix $\hat{C}$ considering the subspace spanned by the eigenvectors corresponding to the largest r eigenvalues.

Assume now that either $\hat{C}$ or X is available and that a new data snapshot x_{k+1} has been computed, causing a change in the signal subspace. The estimate for the signal subspace is then updated incorporating the new information contained in x_{k+1}. This process is frequently called *"subspace tracking"*, referring to the updating of an estimate for the time–varying vector subspace.

A rank–one updating of the covariance matrix $\hat{C}$ and the new snapshot x_{k+1} is involved in each step of the subspace tracking algorithms. Then a new rank–r approximation of the covariance matrix $\hat{C}$ is obtained from the rank–one modification removing from its dyadic decomposition the component corresponding to the largest $(r+1)$th eigenvalue.

Let us shortly describe how the EVD–OPERA algorithm for subspace tracking works.

1) The EVD–OPERA algorithm: Let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive semidefinite with $\mathrm{rank}(A) = r$. The matrix A can be written as

$$A = \sum_{i=1}^{r} \lambda_i v_i v_i^H = V \Lambda V^H, \tag{2}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$ are its positive eigenvalues, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r)$, and $V = [v_1, v_2, \ldots, v_r]$ the matrix of the corresponding eigenvectors, such that $V^H V = I_r$, with I_r the identity matrix of order r. Let $x \in \mathbb{C}^n$ be the new snapshot and consider the rank–one update

$$\hat{A} = A + x x^H.$$

Then

$$x = V\alpha + \alpha_{r+1} v_{r+1}, \tag{3}$$

with

$$\alpha = \left[\alpha_1, \alpha_2 \ldots, \alpha_r\right]^H = V^H x, \quad \alpha_{r+1} = \|(I_n - VV^H)x\|_2,$$

$$v_{r+1} = \frac{(I_n - VV^H)x}{\alpha_{r+1}}, \quad V^H v_{r+1} = 0.$$

Therefore

$$\hat{A} = [V \mid v_{r+1}] \left(\left[\begin{array}{c|c} \Lambda & \\ \hline & 0 \end{array}\right] + \begin{bmatrix} \alpha \\ \alpha_{r+1} \end{bmatrix} \begin{bmatrix} \alpha^H & \alpha_{r+1}^H \end{bmatrix} \right) [V \mid v_{r+1}]^H.$$

Let

$$\tilde{A} = \left[\begin{array}{c|c} \Lambda & \\ \hline & 0 \end{array}\right] + \begin{bmatrix} \alpha \\ \alpha_{r+1} \end{bmatrix} \begin{bmatrix} \alpha^H & \alpha_{r+1}^H \end{bmatrix} \tag{4}$$

and let

$$\tilde{A} = \sum_{i=1}^{r+1} \tilde{\lambda}_i \tilde{v}_i \tilde{v}_i^H$$

be the dyadic decomposition of $\tilde{A}$, with $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{r+1} \geq 0$ its positive eigenvalues and corresponding eigenvectors $\tilde{v}_i$, $i = 1, \ldots, r+1$.

The dyadic decomposition of $\hat{A}$ is given by

$$\hat{A} = \sum_{i=1}^{r+1} \hat{\lambda}_i \hat{v}_i \hat{v}_i^H$$

with

$$\hat{\lambda}_i = \tilde{\lambda}_i, \quad \hat{v}_i = [V \mid v_{r+1}] \tilde{v}_i, \quad i = 1, \ldots, r+1.$$

Then the best rank r approximation matrix to $\hat{A}$ is obtained by dropping the term $\hat{\lambda}_{r+1}\hat{v}_{r+1}\hat{v}_{r+1}^H$ from $\hat{A}$ [6] and setting $V = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_r]$.

The most expensive part of one updating step of the EVD–OPERA algorithm is the computation of $\hat{v}_i, i = 1, \ldots, r+1$, requiring $2nr^2$ operations. The decomposition of x in (3) can be accomplished in $2nr$ floating point operations. Moreover, the eigendecomposition (4) can be computed in a stable way with $O(c_1 r^2)$ operations, with c_1 a constant of moderate size [2], [7]. Therefore, the overall computational complexity is $(4n + c_1)r^2$.

In the next section we describe the new algorithm that tracks an orthogonal subspace approximating the original signal subspace, with $6nr + 15r^2$ computational complexity.

III. TRIDIAGONAL UPDATING ESTIMATION FOR SIGNAL SUBSPACE TRACKING

The important information in the EVD–OPERA algorithm is given by the computed orthogonal signal subspace V. In this section we propose a fast algorithm for computing an orthogonal subspace approximating the original signal subspace, based on the the EVD–OPERA–algorithm.

The role of the diagonal matrix Λ in the EVD–OPERA algorithm is played by a Hermitian tridiagonal matrix T in the new algorithm. In fact, the algorithm is based on *T*ridiagonal *R*ank–one *U*pdating based estimation for signal *S*ubspace *T*racking. Therefore the algorithm will be called *TRUST*.

Given the estimated covariance matrix (1) the initialization can be either accomplished by a nontrivial truncated reduction of the matrix of the Hermitian rank–r matrix (2) into a Hermitian tridiagonal one by Householder transformations [6],

$$A = UTU^T, \tag{5}$$

with $U \in \mathbb{C}^{n\times r}$ unitary and

$$T = \begin{bmatrix} \gamma_1 & \beta_1^H & & & \\ \beta_1 & \gamma_2 & \beta_2^H & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \gamma_{r-1} & \beta_{r-1}^H \\ & & & \beta_{r-1} & \gamma_r \end{bmatrix},$$

or simply computing the spectral decomposition (2) of the matrix A. Due to the truncation of the Householder reduction after $r-1$ steps, the latter initialization procedure turns out to be more reliable.

We observe that the matrices V and U in (2) and (5) span the same signal subspace. The problem now is to compute the new signal subspace of the rank–one modification of A,

$$\hat{A} = A + xx^H,$$

where $x \in \mathbb{C}^n$ is the new snapshot.

The TRUST algorithm is made by the following steps.

Step 1. The new snapshot x is decomposed into its projection onto U and on its projection onto the orthogonal complement $U^\perp$,

$$x = U\hat{\alpha} + \hat{\alpha}_{r+1}u_{r+1},$$

with

$$\hat{\alpha} = \left[\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_r\right]^H = U^H x, \quad \hat{\alpha}_{r+1} = \|(I_n - UU^H)x\|_2,$$

$$u_{r+1} = \frac{(I_n - UU^H)x}{\hat{\alpha}_{r+1}}, \quad U^H u_{r+1} = 0.$$

The problem is transformed into a similar rank–one modification one,

$$\begin{aligned}\hat{A} &= A + x_{r+1}x_{r+1}^H \\ &= [U|u_{r+1}]\left(\left[\begin{array}{c|c} T & \\ \hline & 0 \end{array}\right] + \begin{bmatrix} \hat{\alpha} \\ \hat{\alpha}_{r+1} \end{bmatrix} \left[\hat{\alpha}^H, \hat{\alpha}_{r+1}^H\right]\right)[U|u_{r+1}]^H.\end{aligned}$$

This step is similar to the one of the EVD–OPERA algorithm, requiring the same number of floating point operations, i.e., $2rn$ floating point operations.

Step 2. Reduction of the Hermitian rank–one modification of the matrix T, into a similar Hermitian pentadiagonal matrix S,

$$S = Q_1\left(\left[\begin{array}{c|c} T & \\ \hline & 0 \end{array}\right] + \begin{bmatrix} \hat{\alpha} \\ \hat{\alpha}_{r+1} \end{bmatrix} \begin{bmatrix} \hat{\alpha}^H & \hat{\alpha}_{r+1}^H \end{bmatrix}\right)Q_1^H. \tag{6}$$

This reduction can be accomplished by an algorithm similar to those used for reducing a diagonal plus a Hermitian rank–one matrix into a similar Hermitian tridiagonal one [15], [21]. The reduction, for a matrix of order 8, is depicted in Fig. 1.

To annihilate the entries of the rank–one matrix and to chase the bulges during the construction of the pentadiagonal matrix, $r/2(r/2-1)$ Givens rotations are used. Therefore, the computational complexity of this step is $8r^2$. We observe that the unitary matrix Q_1 in (6), given by the product of the $r/2(r/2-1)$ Givens rotations, is not explicitly computed. The Givens coefficients are stored, requiring $O(r^2)$ memory and used in step 4 to update the signal subspace.

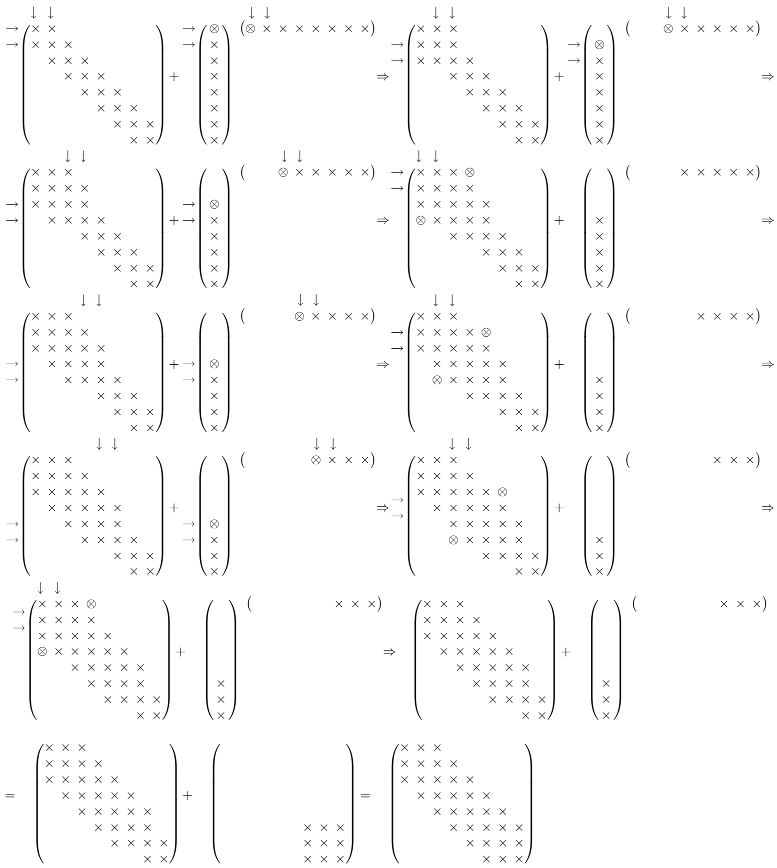

Fig. 1. Description of the unitary transformation used to reduce a Hermitian tridiagonal matrix plus a Hermitian rank–one matrix into a Hermitian tridiagonal one.

Step 3. Reduction of S into a similar Hermitian tridiagonal one $\hat{T}$,

$$\hat{T} = Q_2 S Q_2^H. \tag{7}$$

The reduction of a Hermitian pentadiagonal matrix into a similar Hermitian tridiagonal one can be accomplished by using standard bulge–chasing techniques [17]. Also this step is accomplished by using $r/2(r/2-1)$ Givens rotations with $19/2r^2$ computational complexity. Again, also in this case the unitary matrix Q_2 in (7) is not explicitly computed. The Givens coefficients are stored, requiring $O(r^2)$ memory and used in step 4 to update the signal subspace. This reduction is depicted in Fig. 2 for a matrix of order 8.

Step 4. Update the signal subspace.

Since

$$\begin{aligned}\left(\left[\begin{array}{c|c} T & \\ \hline & 0\end{array}\right] + \begin{bmatrix}\hat{\alpha}\\ \hat{\alpha}_{r+1}\end{bmatrix}\left[\hat{\alpha}^H, \hat{\alpha}_{r+1}^H\right]\right) &= Q_1^H S Q_1 \\ &= Q_1^H Q_2^H \hat{T} Q_2 Q_1\end{aligned}$$

then

$$\hat{A} = [U \mid u_{r+1}] Q_1^H Q_2^H \hat{T} Q_2 Q_1, [U \mid u_{r+1}]^T. \tag{8}$$

Let $\hat{T} = \hat{Z}\hat{\Lambda}\hat{Z}^H$ be the eigendecomposition of $\hat{A}$, with $\hat{\Lambda} = \mathrm{diag}(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{r+1})$, $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_{r+1} \geq 0$. Applying very few steps (from a theoretical point of view only one step is needed) of the implicitly–shifted QR algorithm [6] to the matrix $\hat{T}$ with shift $\hat{\lambda}_{r+1}$, we get

$$\hat{T} = \check{Q}\left[\begin{array}{c|c} \check{T} & \\ \hline & \hat{\lambda}_{r+1}\end{array}\right]\check{Q}^H, \tag{9}$$

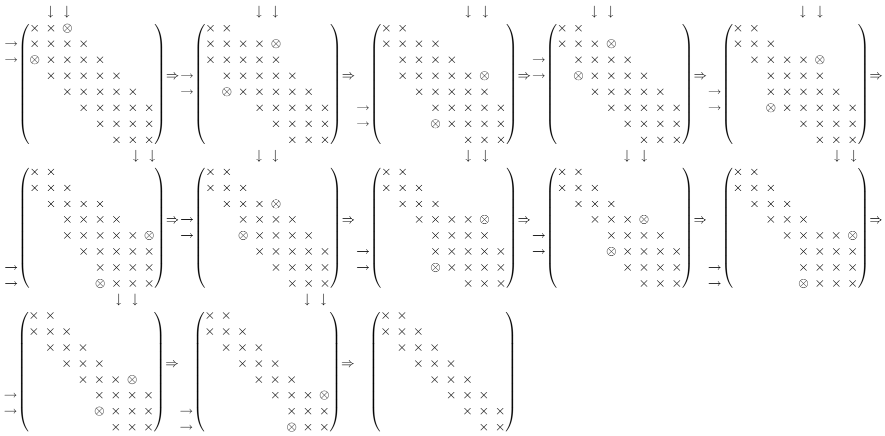

Fig. 2. Description of the unitary transformation used to reduce a Hermitian pentadiagonal matrix into a Hermitian tridiagonal one.

where $\check{T}$ is a Hermitian tridiagonal matrix of order r and $\check{Q}$ is a unitary matrix of order $r+1$. Let us define

$$\Upsilon \equiv \left[\begin{array}{c|c} \check{T} & \\ \hline & \hat{\lambda}_{r+1} \end{array}\right]. \tag{10}$$

Moreover, $\check{q}_{r+1}$, the last column of the matrix $\check{Q}$, is the eigenvector of $\hat{T}$ corresponding to $\hat{\lambda}_{r+1}$, i.e.,

$$\check{q}_{r+1} = \check{Q}e_{r+1} = \hat{Z}e_{r+1}.$$

Let H be the Householder matrix such that

$$Hw_{r+1} = \mp e_{r+1}, \quad w_{r+1} = Q_1^H Q_2^H \check{Q} e_{r+1}. \tag{11}$$

Therefore

$$HQ_1^H Q_2^H \check{Q} = \left[\begin{array}{c|c} W & \\ \hline & \mp 1 \end{array}\right],$$

with $W \in \mathbb{C}^{r\times r}$ unitary. Hence
Taking (9), (10) and (8) into account, it turns out

$$\begin{aligned}
\hat{A} &= [U|u_{r+1}]Q_1^H Q_2^H \hat{T} Q_2 Q_1, [U|u_{r+1}]^T \\
&= [U|u_{r+1}]Q_1^H Q_2^H \check{Q}\Upsilon\check{Q}^H Q_2 Q_1, [U|u_{r+1}]^T \\
&= [U|u_{r+1}]H^H H Q_1^H Q_2^H \check{Q}\Upsilon\check{Q}^H Q_2 Q_1 H^H H [U|u_{r+1}]^T \\
&= [U|u_{r+1}]H^H \left[\begin{array}{c|c} W & \\ \hline & \mp 1 \end{array}\right] \Upsilon \left[\begin{array}{c|c} W & \\ \hline & \mp 1 \end{array}\right]^H H[U|u_{r+1}]^T.
\end{aligned}$$

Let $\hat{U} \equiv U \mid u_{r+1}]H^H$. Due to the special block tridiagonal structure of Υ, the signal subspace associated to the largest r eigenvalues of $\hat{A}$ is given by the first r columns of $\hat{U}$. Given the matrix H, the latter product is computed with $4nr$ floating point operations.
In the end of this step, we set

$$\begin{aligned} U &\leftarrow \hat{U}(:,1:r), \\ T &\leftarrow \check{T}. \end{aligned}$$

The computation of the smallest eigenvalue $\hat{\lambda}_{r+1}$ of the Hermitian tridiagonal matrix $\hat{T}$ can be done in several ways with $O(r)$ floating point operations. For our purposes, we have used a modified version of the rational QR–method with Newton shift described in [16].
Once the smallest eigenvalue $\hat{\lambda}_{r+1}$ has been computed, the corresponding eigenvector can be computed in several ways [5], [6], [14], [22], [23]. For the implementation of the TRUST algorithm we have considered the efficient algorithm described in [14], requiring $O(r)$ floating point operations.
Computed the vector $\check{q}_{r+1}$, w_{r+1} is obtained multiplying $\check{q}_{r+1}$ in (11) to the left first by Q_2^H and then by Q_1^H. Since the matrices Q_1 and Q_2 are not explicitly computed, the latter multiplications are performed reconstructing the Givens matrices from the stored Givens coefficients, and multiplying them by $\check{q}_{r+1}$. Hence, the overall complexity of the latter computation is $3r^2$.

Despite a higher number of steps, the TRUST algorithm has a lower computational complexity compared to the EVD–OPERA algorithm. Indeed, as we will see, the most expensive step is the first and the last one, requiring $6rn+O(r^2)$ floating point operations in total.

At first glance, the EVD–OPERA algorithm and the TRUST algorithm seem to be equivalent. Indeed, if only one updating step is considered, the matrices V and U computed by one step of the EVD–OPERA algorithm and the TRUST algorithm, span the same subspace. However, if one more step is considered, i.e., a new snapshot is processed, we come up with two different rank–one modification problems, (diagonal plus rank–one and tridiagonal plus rank-one for the EVD–OPERA algorithm and the TRUST algorithm, respectively), yielding, in the end of the second updating step, two matrices V and U spanning two different subspaces.

The TRUST algorithm has been implemented in `Matlab`[2]. The code can be obtained from the authors upon request.

IV. Numerical experiments

To compare the efficiency of the TRUST algorithm with respect to the EVD–OPERA algorithm, the following examples have been considered. The eigenvalue decomposition (6) in the OPERA algorithm has been computed using the function `eig` of `Matlab`. As previously said, this computation could be performed in a more efficient way considering, e.g., the algorithm described in [7] for computing the eigendecomposition of a Hermitian rank–one modification of a diagonal matrix. For the TRUST algorithm, two different implementations have been considered, differing for the two different initializations described in section III. To measure the number of floating point operations the algorithms have been run in the `Matlab 5.3` environment, neglecting the cost of the initialization step for the algorithms considered.

Example 1: Let $V\in\mathbb{R}^{100\times 20}$ and $W\in\mathbb{R}^{20\times 100}$ be full rank random matrices generated by the `matlab` function `randn`. Let $X=VW+\tau Z$, with $Z\in\mathbb{R}^{100\times 100}$ be a random matrix generated by the `matlab` function `randn` and τ a parameter, assuming different values, that gives an indication of the level of noise, i.e., of the distance from the subspace spanned by the columns of V and that one spanned by the columns of X. If $\tau=0$, V and X span the same subspace of $\mathbb{R}^{100}$. Let V_1, V_2, and V_3 be the orthogonal subspaces tracked by the EVD–OPERA algorithm (Alg1), by the TRUST algorithm with eigenvalue initialization (Alg 2), by the TRUST algorithm with tridiagonal initialization (Alg3), respectively. For the initialization step the first 20 columns of X have been considered, i.e., $\hat{C}=\frac{1}{20}X(:,1:20)X(:,1:20)^H$. As a measure of the accuracy of the subspaces tracked, we have computed

$$\|V^H(I-V_iV_i^H)\|_2,\quad i=1,2,3.$$

The results are reported in table 1. Since the complexity is quite independent from the level of the noise, we report only the number of flops needed to track the subspaces by Alg1, Alg2 and Alg3 for $\tau=1e-2$. It can be noticed that the accuracy of

τ	$\|V^H(I-V_1V_1^H)\|_2$	$\|V^H(I-V_2V_2^H)\|_2$	$\|V^H(I-V_3V_3^H)\|_2$
1.0e-02	1.6826e-02	4.2094e-02	2.1759e+00
1.0e-04	1.6825e-04	5.4624e-04	2.3606e-02
1.0e-06	1.6825e-06	4.5224e-06	2.1609e-04
1.0e-08	1.6825e-08	6.5214e-08	2.4493e-06
1.0e-10	1.6825e-10	4.2234e-10	2.3454e-08
1.0e-12	1.6876e-12	4.7211e-12	7.8048e-10
1.0e-14	5.1938e-14	8.8494e-14	7.4705e-10
	# flops Alg1	# flops Alg2	# flops Alg3
	14865653	2737821	2769177

TABLE I

Levels of orthogonality between the subspace V and the tracked subspaces $V_i, i=1,\ldots,3$.

the subspaces tracked by the three methods is comparable and proportional to the amount of the noise introduced (the term τZ) although there is a loss of accuracy in the TRUST algorithm with the tridiagonal initialization.

Example 2: This example is very similar to the previous one. Let $V\in\mathbb{R}^{200\times 40}$ and $W\in\mathbb{R}^{40\times 200}$ be full rank random matrices generated by the `matlab` function `randn`. Let $X=VW+\tau Z$, with $Z\in\mathbb{R}^{200\times 200}$ be a random matrix generated by the `matlab` function `randn` and τ as in the previous example. Let V_1, V_2, and V_3 be the orthogonal subspaces tracked by the EVD–OPERA algorithm (Alg1), by the TRUST algorithm with eigenvalue initialization (Alg 2), by the TRUST algorithm with tridiagonal initialization (Alg3), respectively. For the initialization step the first 40 column of X have been considered. The results are reported in table 2.

V. Conclusions

A fast algorithm that computes the orthogonal subspace tracked by the EVD–OPERA algorithm is described in this paper. Its complexity is $6nr+15r^2$ (the complexity of the EVD–OPERA algorithm is $2nr^2+O(r^2)$), where n is the input vector dimension and r ($n\gg r$) is the desired number of eigencomponents. The algorithm is shown to be efficient and stable.

[2]`Matlab` is a trademark of the MathWorks Inc.

τ	$\|V^H(I-V_1V_1^H)\|_2$	$\|V^H(I-V_2V_2^H)\|_2$	$\|V^H(I-V_3V_3^H)\|_2$
1.0e-02	1.7905e-02	1.2399e-01	6.5583e+00
1.0e-04	1.8198e-04	9.7354e-04	1.4746e-02
1.0e-06	1.7585e-06	1.1072e-05	2.8420e-05
1.0e-08	1.8248e-08	1.3242e-07	4.7096e-07
1.0e-10	1.6180e-10	1.2536e-09	8.3852e-09
1.0e-12	1.8049e-12	5.1923e-11	9.7479e-08
1.0e-14	1.1956e-13	4.4641e-12	1.1920e-09
	# flops Alg1	# flops Alg2	# flops Alg3
	214848759	21211667	21281585

TABLE II

LEVELS OF ORTHOGONALITY BETWEEN THE SUBSPACE V AND THE TRACKED SUBSPACES $V_i, i=1,\ldots,3$.

REFERENCES

[1] S. ATTALLAH AND K. ABED-MERAIM, *Fast algorithms for subspace tracking*, IEEE Signal Process. Lett., vol. 8 (7): 203-206 JUL 2001

[2] C.F. BORGES AND W.B. GRAGG, *A parallel divide & conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem*, in Numerical Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan and R.S. Varga, eds., de Gruyter, Berlin, 1993, pp.11–29.

[3] Y. CHAHLAOUI, K. GALLIVAN, AND P. VAN DOOREN, *Recursive calculation of dominant singular subspaces,* SIAM J. Matrix Anal. Appl., 25(2):445–463, 2004.

[4] B. CHAMPAGNE AND Q.-G. LIU, *Plane rotation-based EVD updating schemes for efficient subspace tracking*, IEEE Transactions on Signal Processing, vol. 46, pp. 1886-1900, July 1998.

[5] I.S. DHILLON, B.N. PARLETT, *Orthogonal Eigenvectors and Relative Gaps*, SIAM J. Matrix Anal. Appl., 25(3):858–899, 2004.

[6] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[7] M. GU AND S.C. EISENSTAT, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15(4): 1266-1276, 1994.

[8] T. GUSTAFSSON AND C.S. MACINNES, *A class of subspace tracking algorithms based on approximation of the noise–subspace*, IEEE Trans. Signal Process., 48(11): 3231-3235, Nov 2000.

[9] Y. HUA, Y. XIANG, T. CHEN, K. ABED-MERAIM, AND Y. MIAO, *A new look at the power method for fast subspace tracking*, Digital Signal Processing, vol. 9:297–314, Oct. 1999.

[10] C.S. MACINNES, *Fast, Accurate subspace tracking using operator restriction analysis*, in Proc. of IEEE ICASSP, 1998, pp. 1357-1360.

[11] C.S. MACINNES, *Source localization using subspace estimation and spatial filtering,* IEEE J. Ocean. Eng., vol. 29 (2): 488-497 APR 2004.

[12] C.S. MACINNES, *DOA estimation of nonstationary coherent signals using signal vector randomization*, IEEE Trans. Signal Process., vol. 46 (6): 1744-1749 JUN 1998.

[13] C.S. MACINNES AND R. VACCARO, *Tracking directions-of-arrival with invariant subspace updating*, Signal Process., 50 (1-2): 137-150 APR 1996.

[14] N. MASTRONARDI, M. VAN BAREL, E. VAN CAMP, AND R. VANDEBRIL, *On computing the eigenvectors of a class of structured matrices*, J. Comput. Appl. Math., to appear.

[15] H. PARK AND S. VAN HUFFEL, *Two–way bidiagonalization scheme for downdating the singular value decomposition*, Lin. Alg. Appl., vol. 222, 1995, pp. 23-39.

[16] C. REINSCH AND F.L. BAUER, *Rational QR Transformation with Newton Shift for Symmetric Tridiagonal Matrices*, Numer. Math., vol. 11, pp. 264–272, 1968.

[17] G.W. STEWART, *Matrix Algorithms. Volume II: Eigensystems*, SIAM, Philadelphia, 2001.

[18] P. STOICA, AND K.C. SHARMAN, *A novel eigenanalysis method for direction estimation*, Radar and Signal Processing IEE Proceedings, vol. 137 (1): pp. 19–26, Feb 1990.

[19] E.C. REAL, D.W. TUFTS AND J.W. COOLEY, *Two algorithms for fast approximate subspace tracking*, IEEE Trans. Signal Process., vol. 47:7, pp. 1936–1945, July 1999.

[20] D.W. TUFTS, E.C. REAL AND J.W. COOLEY, *Fast approximate subspace tracking (FAST)*, ICASSP-97, IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997, vol.1, pp. 547–550.

[21] S. VAN HUFFEL AND H. PARK, *Parallel Tri– and Bi–Diagonalization of Bordered Bidiagonal Matrices*, Parallel Computing 20(8): 1107–1128, 1994.

[22] S. VAN HUFFEL, J. VANDEWALLE AND A. HAEGEMANS, *An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values*, J. Comput. Appl. Math., vol. 19, 1987, pp. 313–330.

[23] S. VAN HUFFEL, *Iterative algorithms for computing the singular subspace of a matrix associated with its smallest singular values*, Lin. Alg. Appl., vol. 154-156, 1991, pp. 675–709.

[24] B. YANG, *Projection approximation subspace tracking*, IEEE Trans. Signal Process., vol. 43, pp. 95-107, Jan. 1995.