ACCURATE SUBSPACE TRACKING ALGORITHMS BASED ON CROSS-SPACE PROPERTIES

Philippe A. Pango and Benoît Champagne

INRS-Télécommunications, 16 Place du Commerce
Verdun, Québec, Canada, H3E 1H6
pango@inrs-telecom.uquebec.ca

ABSTRACT

In this paper, we analyse the issue of efficiently using Givens rotations to perform a more accurate SVD-based subspace tracking. We propose an alternative type of decomposition which allows a more versatile use of Givens rotations. We also show the direct effect of the latter on the tracking error, and develop a cross-terms cancellation concept which leads to a class of high performance algorithms with very low complexity: $O(N^2)$ if signal and noise subspaces are tracked, $O(Nr)$ if only the signal subspace is tracked, where N is the data vector dimension, and r the number of sources. Comparative simulation experiments support the theoretical work.

1. INTRODUCTION

In the application of subspace methods to non-stationnary signals, one often has to compute an approximate singular value decomposition (SVD) of a data matrix of growing dimension, defined recursively as

$$A(k) = \begin{bmatrix} \lambda A(k-1) \\ \mathbf{x}^H(k) \end{bmatrix} \tag{1}$$

where k is the time index, $0 < \lambda < 1$ is the forgetting factor, and $\mathbf{x}(k) \in \mathbb{C}^N$ is the incoming measurement vector. Whereas an exact SVD requires $O(N^3)$ operations, Moonen [1] has proposed an algorithm with a complexity of $O(N^2)$ operations per update which provides a decomposition

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

where $V(k)$ is an $N \times N$ unitary matrix, $U(k)$ is an $k \times N$ matrix with orthonormal columns and $\Sigma(k)$ is an upper-triangular, almost diagonal matrix. This decomposition is computed in two steps: (*i*) a *QR step* achieved by N Givens rotations which annihilate the entries of the incoming vector, and (*ii*), a Jacobi-type *SVD step* using $N-1$ Givens rotations along the second diagonal of $\Sigma(k)$ to reduce its off-norm. Kavcic [2] introduced a noise sphericalized version of this algorithm, the Noise Average SVD (NASVD), based on the spherical EVD update from DeGroat [3] and Stewart's URV decomposition [4]; it reduces the complexity to $O(Nr)$ by tracking only an r-dimensional signal subspace. Based on the same structure but with a different *SVD step*, Rabideau [5] has proposed an algorithm of similar complexity, the RO-FST, which considerably improves the tracking performance.

All the above algorithms use Givens rotations in a way which is not proved to be optimal. In this paper, we first propose an alternative type of approximate SVD and then use it to investigate the effect of Givens rotations on the updating process. Using an innovative convergence strategy, new SVD updating algorithms are derived for both signal-and-noise and signal-only subspace tracking. Finally, comparative computer experiments are used to validate the proposed approach.

This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada

2. CONVERGENCE STRATEGY

2.1. A new type of decomposition

Convergence to the exact SVD is closely related to the ability of the algorithm to diagonalise $\Sigma(k)$ faster and to rotate the approximate singular vectors in the directions of the exact ones. The Givens rotation can achieve both tasks if it is judiciously used. We recall that a Givens rotation matrix $G_\theta^{i|j}$ differs from an identity matrix only at four entries: $G_\theta^{i|j}(i,i) = c$, $G_\theta^{i|j}(i,j) = s$, $G_\theta^{i|j}(j,i) = -s$ and $G_\theta^{i|j}(j,j) = c$, where c and s are the cosine and sine of the angle θ.

Suppose that the approximate SVD of $A(k-1)$, defined as in (2), is known and that we want to compute the SVD of (1). Once we have applied the *QR step* to eliminate the new row of $A(k)$ [1], the improvement of the diagonalisation of $\Sigma(k)$ is achieved by applying a series of Givens rotations. This diagonalisation step, sometimes called *SVD step* or *refinement*, exists in different flavours in the litterature. On one hand, RO-FST uses two series of Givens rotations to zero alternatively the last column and the last row of $\Sigma(k)$. On the other hand, [1, 2] apply a series of pairs of Givens rotations on both sides of $\Sigma(k-1)$ as in

$$\Sigma(k) \leftarrow [G_\phi^{i|j}]^T \Sigma(k) G_\theta^{i|j}, \tag{3}$$

where $j = i+1$; and for each of these pairs, the rotation angles are computed so as to zero the (i,j) and (j,i) entries. We call the (i,j) entry the *pivot* of the rotation.

Both methods are designed so as to maintain $\Sigma(k)$ in an upper-triangular structure and to reduce the off-norm of $\Sigma(k)$. Yet, a priori, it is not proved that applying rotations

as these methods do is the optimal way to reduce the off-norm of $\Sigma(k)$ under the constraint of a fixed number of Givens rotations. We also note that the most important attribute for $\Sigma(k)$ is not to be upper-triangular, but to have the smallest possible off-norm, since much of the subspace approximation error is due to the off-diagonal terms of $\Sigma(k)$.

It can be verified that in (3), the off-norm of $\Sigma(k)$ is reduced as

$$off[\Sigma(k)] \leftarrow off[\Sigma(k)] - [\sigma(i,j)^2 + \sigma(j,i)^2] \qquad (4)$$

where $\sigma(i,j)$ is the pivot entry of both rotations. So, in Moonen-type algorithms, the diagonalisation would be efficient only if the $(i, i+1)$ entries remain the largest off-diagonal entries (in absolute values) throughout the diagonalisation process. The larger $\sigma(i,j)$ and $\sigma(j,i)$, the better the diagonalisation.

This observation immediately leads us to formulate an algorithm which tracks the position of the maximum off-diagonal entry of $\Sigma(k)$ before each pair of rotations, so that rotations are always applied to the largest pivots. By achieving a better off-norm reduction, this "Maximum-Search" algorithm (MS) provides a more accurate tracking, when compared to Moonen's, NASVD, ROSA, and even RO-FST algorithm, as will be shown in our simulations in Section 4. Since off-diagonal maximums could be located anywhere in $\Sigma(k)$, rotations will be applied in almost random locations and $\Sigma(k)$ will not remain upper-triangular.

Accordingly, the good performance of the MS algorithm, as observed experimentally, shows that the upper-triangular structure used in previous algorithms is not required. To allow the possibility of applying Givens rotations anywhere in $\Sigma(k)$, we need to consider a new type of approximate SVD, still expressed as in (2), but where $\Sigma(k)$ is now almost diagonal and not specifically upper-triangular. We refer to this decomposition as an UXV.

2.2. Block diagonalisation

Since the search for a single maximum adds a complexity of $N^2 \log N^2$ to the overall algorithm complexity, we need to find some practical way to achieve a better subspace tracking and a good diagonalisation without tracking the largest off-diagonal entry's position before each rotation.

After the *QR step*, the intermediate decomposition is as in (2) (now using time index k) and needs to be refined. Let us define the following submatrices:

$$U(k) = [U_S(k)|U_N(k)], \quad V(k) = [V_S(k)|V_N(k)], \qquad (5)$$

and the almost diagonal singular value matrix

$$\Sigma(k) = \begin{bmatrix} \Sigma_S(k) & \Sigma_{SN}(k) \\ \Sigma_{NS}(k) & \Sigma_N(k) \end{bmatrix}. \qquad (6)$$

The main diagonal of $\Sigma(k)$ contains the approximated singular values which are supposed to be in decreasing order: $\sigma(1) > \sigma(2) > \cdots > \sigma(N)$. The singular vectors have been separated in two sets corresponding to the r largest and $N-r$ smallest singular values; the signal subspace is defined as the column-span of $V_S(k)$. At this point, we have

$$A(k) \cong U_S(k)\Sigma_S(k)V_S(k)^H + U_N(k)\Sigma_N(k)V_N(k)^H + A_{SN}(k) \qquad (7)$$

where

$$A_{SN}(k) = U_S(k)\Sigma_{SN}(k)V_N(k)^H + U_N(k)\Sigma_{NS}(k)V_S(k)^H \qquad (8)$$

The exact decomposition at time k can be written as

$$A(k) = U_S^o(k)\Sigma_S^o(k)V_S^o(k)^H + U_N^o(k)\Sigma_N^o(k)V_N^o(k)^H \qquad (9)$$

where Σ_S^o and Σ_N^o are diagonal submatrices, containing the exact signal and noise singular values. By comparing (7) and (9), we see that the approximate expression has the correct form of an SVD when $A_{SN}(k) = \mathbf{0}$, which is achieved when the off-diagonal blocks are null, i.e. $\|\Sigma_{SN}\| = \|\Sigma_{NS}\| = 0$. We note that the approximate subspaces are not yet identical to the exact ones since the submatrices Σ_S and Σ_N are not diagonal; nevertheless, the block diagonalisation helps to cancel the cross-interaction between subspaces of different nature and improves the subspace information.

To substantiate this affirmation, we will now analyse the direct effect of block diagonalisation on the tracking error. Our goal here is to investigate the effect of each Givens rotation on the signal subspace projector $V_S(k)V_S(k)^H$ obtained from the tracking algorithm. Recall that this type of 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 subspaces

$$TE(k) = \|V_S^o(k)V_S^o(k)^H - V_S(k)V_S(k)^H\|_2 \qquad (10)$$

where "o" stands for "exact decomposition". Similarly, define the *time variation* of the updating process, $\Delta V_S(k)$, as the distance between the approximate signal subspaces at time $k-1$ and k. In Jacobi-type subspace tracking algorithms, the refinement step transforms the right singular vectors according to

$$V(k) \leftarrow V(k) \prod_{l=1}^{N-1} G_{\theta_l}^{i_l|j_l}, \qquad (11)$$

where (i_l, j_l) represents the pivot position and θ_l the angle of the l^{th} Givens rotation. Each Givens rotation affects only the i_l^{th} and the j_l^{th} right singular vector. Let us note that in Moonen-type algorithms, we have $j_l = i_l + 1$. Finally, we define Δ_l, a measure of the partial alteration of $V_S(k-1)V_S(k-1)^H$ that is made by the l^{th} rotation, as the distance between the approximated signal subspaces before and after the l^{th} rotation in (11).

Whereas $\Delta V_S(k) = 0$ means either perfect convergence or a useless algorithm, $\Delta_l = 0$ is the result of an unnecessary Givens rotation. Furthermore, since each rotation in (11) helps to improve the diagonalisation of $\Sigma(k)$ (consequence of (4)) 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 l^{th} rotation to $\Delta V_S(k)$. Thus, our viewpoint here is that, like the MS algorithm which reaches greater performance by performing a better diagonalisation at each rotation, Δ_l should be as large as possible.

Temporarily denote by $U(l-1)\Sigma(l-1)V(l-1)^H$ the approximate decomposition before the l^{th} 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 (5) and (6) with time index k replaced by rotation index $l-1$. There are three possible choices for the location of the 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):

1: $r+1 \leq i_l < j_l \leq N$: the pivot is located in $\Sigma_N(l-1)$ and the rotation affects only two noise singular vectors. Since the signal subspace $V_S(l-1)$ is unchanged, we have $\Delta_l = 0$.

2: $1 \leq i_l < j_l \leq r$: the pivot is located in $\Sigma_S(l-1)$ and the rotation affects two vectors of the signal subspace. The modification of the signal subspace is

$$V_S(l) = V_S(l-1)G^{ul} + V_N(l-1)G^{ll} \qquad (12)$$

where G^{ul} and G^{ll} denote the $r \times r$ upper-left and $(N-r) \times r$ lower-left part of $G_{\theta_l}^{i_l|j_l}$. Here, $G^{ll} = \mathbf{0}$ and G^{ul} is unitary, so, it is easily shown that even if two signal vectors are rotated, we still have $\Delta_l = 0$.

3: $1 \leq i_l \leq r$ and $r+1 \leq j_l \leq N$: the pivot is located in $\Sigma_{SN}(l-1)$. In that case, one vector is rotated in each subspace. The signal subspace transformation is still expressed as in (12), 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 $\Delta V_S(k)$.

In summary, during the refinement, each rotation $G_{\theta_l}^{i_l|j_l}$ modifies the product $V_S(k)V_S(k)^H$ and thus contributes to the time variation $\Delta V_S(k)$ if and only if the pivot is chosen in Σ_{SN} (or Σ_{NS}). Then, it follows that locating the pivots in Σ_{SN} guarantees that the MUSIC detector may be improved at each rotation; otherwise, the rotations have no effect on the parameters estimates, since the noise subspace projection $V_N(k)V_N(k)^H = I - V_S(k)V_S(k)^H$ remains unchanged. A further justification is a theorem from Stewart [6], according to which $\|\Sigma_{SN}\|_2$ can be used to compute an upper-bound on the tracking error as follows :

$$TE(k) < ct\|\Sigma_{SN}(k)\|_2 \qquad (13)$$

where ct is a constant. Indeed, each rotation in Σ_{SN} reduces $\|\Sigma_{SN}\|_2$ by annihilating the (i_l, j_l) and (j_l, i_l) cross-terms entries, thus reducing the bound.

Moonen-type algorithms, whose pivots are located exclusively on the second diagonal have only one rotation in Σ_{SN} (the r^{th} one); this explains why they have a lower convergence rate and thus, a tracking error which often saturates at a higher level.

In the case of partial tracking based on noise subspace sphericalisation, Householder transformations are introduced to reduce the complexity of the algorithms. In particular, since the noise vectors are rotated so that the projection of the incoming data vector onto the noise subspace only has a single non-zero component, Σ_{SN} is reduced to a single column. Yet, the argumentation described above is the same: by locating the pivots of the rotation in that column, one takes advantage of the potential contribution of each rotation and, by reducing the norm of that column, one cancels the interaction between signal and noise subspaces.

3. PROPOSED ALGORITHMS

Based on the above considerations, we propose a new family of algorithms for complete and partial subspace tracking: Cross-space Singular Value decomposition (CSVD) and Noise-Average CSVD (NA-CSVD), since they are based on the annihilation of the cross-terms, as represented by Σ_{SN} in (8).

The major difference with the previous algorithms is the type of decomposition which does not restrict $\Sigma(k)$ to be upper-triangular. The basic idea is to locate pivots in Σ_{SN} in order to take advantage of the potential contribution of each rotation. Tables 1 and 2 show only the main steps of CSVD and NA-CSVD, all other steps (initialisation, singular values averaging) being the same.

The CSVD (Table 1) can be initialised with the exact SVD or for example, with $V(0) = I$ and $\Sigma(0) = \mathbf{0}$, where I denotes for the $N \times N$ identity matrix. The QR step consists in N transposed Givens rotations represented by $Q^T(k)$, after which, the remaining elements in the last row are considered small enough to be forced to zero. In the refinement step, we would like to select the largest pivots each time we apply a rotation; however, in order not to increase the complexity, we propose to generate randomly the $N-1$ pivot positions within Σ_{SN}.

In NA-CSVD (Table 2), the QR step almost zero the projection of the incoming vector on the tracked signal subspace and the noise vector $\mathbf{v_N}$ (see [3] for details); the cross-terms are contained in a vector because of the introduction of a Householder transformation. Accordingly, in the refinement step, the r pivots are directly chosen from the first entry down to the r^{th} entry of the last column of $\Sigma(k)$.

Here, the concept of cross-terms cancellation leads to a type of RO-FST structure and provides an explanation of its good performance. However, our approach is based on a different type of decomposition and both algorithms have different refinement structures.

The complexity of CSVD and NA-CSVD are respectively $O(N^2)$ and $O(Nr)$.

4. EXPERIMENTAL RESULTS

In all our experiments, we estimate the direction of arrival (DOA) of r incident plane waves on a receiving linear array of N sensors. The i^{th} entry of the measurement vector $\mathbf{x}(k)$ represents the combined effect of all the sources at the i^{th} sensor, i.e. $x_i(k) = \sum_{l=1}^{r} A_l(k)e^{j(i-1)\omega_l(k)} + n_i(k)$, $i = 1, \cdots, r$, where $\omega_l(k)$ and $A_l(k)$ are the electrical angle and the complex amplitude of the l^{th} source and $n_i(k)$ is an additive noise component. $n_i(k)$ and $A_l(k)$ are modelled as complex circular Gaussian variables; the variance of $n_i(k)$ is set to 1, and the complex amplitudes $A_l(k)$ correspond to each of the source's signal-to-noise ratio SNR_l.

In the first experiment, we compare complete tracking algorithms: CSVD, MS (the "maximum-search" algorithm described in section 2) and Moonen's algorithm. The simulation parameters are: $r = 2$ fixed sources with DOA of $40°$ and $55°$; $N = 8$ sensors; $SNR = 5dB$; $\lambda = 0.99$. All the algorithms are initialised with the exact SVD at $k = 1$.

Fig. 1(a) shows the initial convergence averaged over 20 experiments. The MS algorithm is the closest to the

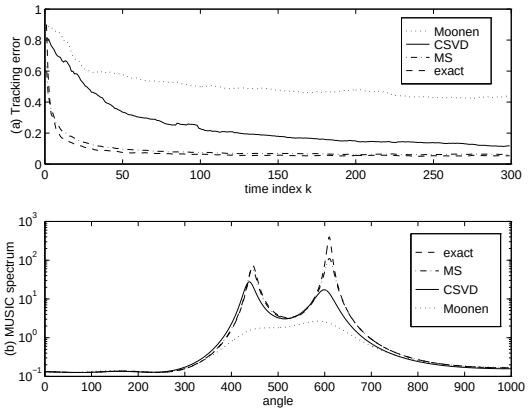

Figure 1: Complete tracking : signal subspace error and MUSIC spectrum for CSVD, MS, Moonen and exact SVD algorithms.

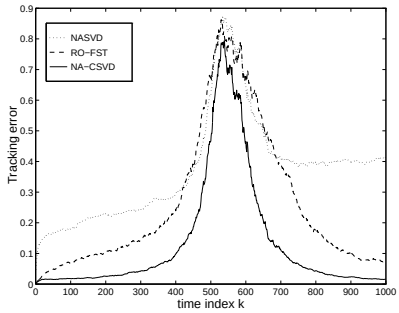

Figure 2: Partial tracking : signal subspace error for two closed moving sources : NA-CSVD, NASVD and RO-FST

exact SVD, confirming that the upper-triangular structure is not required. Next to MS algorithm is the CSVD. Fig. 1(b) shows the MUSIC spectrums after 300 samples for one experiment.

The second simulation tests various partial tracking algorithms: NASVD, RO-FST (one refinement) and NA-CSVD. We limited our simulations to those three since [7] offers numerous comparison tests between RO-FST and other popular algorithms. The simulation parameters are: $r = 2$ crossing sources with DOA's given by $(20 + 0.01k)^\circ$ and $(30 - 0.01k)^\circ$ and crossing at $k = 500$; $N = 8$; $\lambda = 0.99$ and $SNR = 5dB$.

Fig. 2 shows the distance between estimated and true signal subspace, averaged over 20 experiments. It shows that all algorithms face a stronger difficulty when sources meet, since one source is lost. The center of the figure informs on their respective ability to separate close sources, while the sides indicate their ability to track moving sources. Here, NA-CSVD offers an improvement compared to RO-FST and appears to be less sensible to non-stationnarities.

Generally, the algorithms based on cross-terms cancellation (CSVD, NA-CSVD, RO-FST) have a higher performance compared to other algorithms with the same complexity.

Table 1: CSVD algorithm

Step	Operation
QR step	$V(k) \leftarrow V(k-1)$ $\begin{bmatrix} \Sigma(k) \\ \cdots \end{bmatrix} \leftarrow Q^T(k) \begin{bmatrix} \lambda\Sigma(k-1) \\ \mathbf{x}^H(k)V(k) \end{bmatrix}$
refine-ment	for $l = 1 : N-1$ choose (i_l, j_l) randomly in Σ_{SN} $\Sigma(k) \leftarrow [G_\phi^{i_l\|j_l}]^T \Sigma(k) G_\theta^{i_l\|j_l}$ $V(k) \leftarrow V(k) G_\theta^{i_l\|j_l}$ end

Table 2: NA-CSVD algorithm

Step	Operation
spherica-lisation	$V_S(k) \leftarrow V_S(k-1)$ $\mathbf{v}_N \leftarrow f(\mathbf{x}(k), V_S(k))$ $V(k) \leftarrow [V_S(k), \mathbf{v}_N]$
QR step	$\begin{bmatrix} \Sigma(k) \\ \cdots \end{bmatrix} \leftarrow Q^T(k) \begin{bmatrix} \lambda\Sigma(k-1) \\ \mathbf{x}^H(k)V(k) \end{bmatrix}$
refine-ment	for $l = 1 : r$ $\Sigma(k) \leftarrow [G_\phi^{l\|L+1}]^T \Sigma(k) G_\theta^{l\|L+1}$ $V(k) \leftarrow V(k) G_\theta^{l\|L+1}$ end

5. REFERENCES

[1] M. Moonen, Van Dooren, J. Vandewalle, "A singular value decomposition algorithm for subspace tracking," *SIAM J. Matrix Anal. Appl., vol. 13, No 4,* pp. 1015-1038, Oct. 1992.

[2] A. Kavcic, B. Yang, "A new efficient subspace tracking based on singular value decomposition," *Proc. IEEE Int. Conf. ASSP*, pp. IV/485-488, 1994.

[3] R. D. Degroat, "Noniterative subspace tracking," *Proc. IEEE Trans. Signal Processing*, vol. 40, No. 3, pp. 571-577, March 1992.

[4] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 1535-1541, 1992.

[5] D. J. Rabideau and A. O. Steinhardt, "Fast subspace tracking", *Proc. Seventh SP Workshop Stat. Signal Array Processing*, pp. 353-356, June 1994.

[6] G. W. Stewart, "Error and perturbation bounds for subspaces associated with certain eigenvalue problems," *SIAM Rev.*, vol. 15, pp.727-764, 1973.

[7] D. J. Rabideau, "Fast, rank adaptive subspace tracking and applications," *IEEE Trans. Signal Processing*, vol. 44, No.9, pp. 2229-2244, Sept. 1996.