CONJUGATE GRADIENT ALGORITHMS FOR MINOR SUBSPACE ANALYSIS

Roland BADEAU, Bertrand DAVID, Gaël RICHARD

Télécom Paris - Département TSI
46 rue Barrault - 75634 PARIS cedex 13 - FRANCE

ABSTRACT

We introduce a conjugate gradient method for estimating and tracking the minor eigenvector of a data correlation matrix. This new algorithm is less computationally demanding and converges faster than other methods derived from the conjugate gradient approach. It can also be applied in the context of minor subspace tracking, as a pre-processing step for the YAST algorithm, in order to enhance its performance. Simulations show that the resulting algorithm converges much faster than existing minor subspace trackers.

Index Terms— Conjugate gradient methods, Minor subspace analysis, Subspace tracking.

1. INTRODUCTION

Fast estimation and tracking of the principal or minor subspace of a sequence of random vectors is a major problem in many applications, such as adaptive filtering and system identification (*e.g.* source localization, spectral analysis) [1]. In the literature, it is commonly admitted that minor subspace analysis (MSA) is a more difficult problem than principal subspace analysis (PSA)[1]. In particular, the classical Oja algorithm [2] is known to diverge. Some more robust MSA algorithms have been presented in [3, 4]. However the convergence rate of these algorithms remains much lower than that of the classical PSA techniques. Recently, we presented in [5] a new minor subspace tracker dedicated to time series analysis. This algorithm, referred to as YAST, reaches the lowest complexity found in the literature, and outperforms classical methods in terms of subspace estimation.

Usual MSA techniques are derived from the gradient approach, applied to the minimization of an appropriate cost function. However, the conjugate gradient technique, initially introduced for solving linear systems [6, pp. 520–530], is known to offer a much faster convergence rate. This approach was applied to the computation and tracking of the minor component of a correlation matrix [7–10]. Here we propose a new algorithm for tracking the minor component, inspired by the preconditioned conjugate gradient (PCG) method presented in [11], which converges faster than existing methods. We show that this new algorithm can be used as a pre-processing for the YAST minor subspace tracker, in order to enhance its performance.

The paper is organized as follows. The new conjugate gradient method for computing the minor component of a correlation matrix is introduced in section 2. Section 3 presents an adaptive version of this algorithm, and describes how it can be applied to minor subspace tracking. Simulation results are presented in section 4. Finally, the main conclusions of this paper are summarized in section 5.

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

[1] Note that computing the minor subspace of a correlation matrix can simply be performed by applying PSA techniques to the inverse matrix. However this approach is often disregarded, because of its high complexity. This is why there is so much interest in designing specific algorithms for MSA.

2. COMPUTATION OF THE MINOR EIGENVECTOR

2.1. Constrained descent methods

The Rayleigh quotient of an $n \times n$ correlation matrix C_{xx} is[2]

$$J : w \in \mathbb{C}^n \mapsto w^H C_{xx} w \Big/ \|w\|^2 . \tag{1}$$

It is well known that the minimal value of the Rayleigh quotient is the lowest eigenvalue λ of the matrix C_{xx}, which is reached when w is an eigenvector of C_{xx} associated to λ. Therefore a possible approach for computing w consists in recursively minimizing this quotient. This optimization can be carried out by means of usual descent methods, such as the gradient (also known as *steepest descent*) or the conjugate gradient methods. An interesting property of the Rayleigh quotient is that its critical points, at which the gradient is zero, are either unstable saddle points or the global minimum or maximum. Therefore the descent methods are usually guaranteed to find the global minimum, unless the initial guess for the eigenvector is deficient in the direction corresponding to the lowest eigenvalue[3].

From a given vector $w(0)$, the descent methods compute a series of vectors $\{w(t)\}_{t\in\mathbb{N}}$ which converges to w. Here we focus on constrained implementations of these methods, where the vectors $w(t)$ belong to the unit sphere ($\|w(t)\| = 1$). The steepest descent approach consists in looking for the new unitary vector $w(t)$ in the subspace spanned by the previous unitary vector $w(t-1)$ and the gradient $\nabla J(w(t-1))$, which is denoted $\nabla J(t-1)$ below. Differentiating equation (1) yields

$$\nabla J(t-1) = 2\left(C_{xx}\, w(t-1) - \lambda(t-1) w(t-1)\right), \tag{2}$$

where $\lambda(t-1) = J(w(t-1))$. Since $\nabla J(t-1)$ and $w(t-1)$ are orthogonal, the unit vector $w(t)$ can be written in the form

$$w(t) = w(t-1)\cos(\theta) - g(t-1)\sin(\theta),$$

where $g(t-1) = \frac{\nabla J(t-1)}{\|\nabla J(t-1)\|}$, and θ is the angular step of the gradient method. The optimal step is that which minimizes $J(w(t))$ with respect to θ, which is equivalent to the algorithm proposed in [10]. Compared to the steepest descent approach, our conjugate gradient method additionally takes the previous descent direction into account. Thus the new vector $w(t)$ is searched in the subspace spanned by the previous vector $w(t-1)$, the normalized gradient $g(t-1)$, and the unitary vector $p(t-1)$, tangent to the unit sphere and orthogonal

[2] The row vector w^H is the conjugate transpose of the column vector w.

[3] In practice however, this singular case is never observed because of rounding errors due to the finite machine precision.

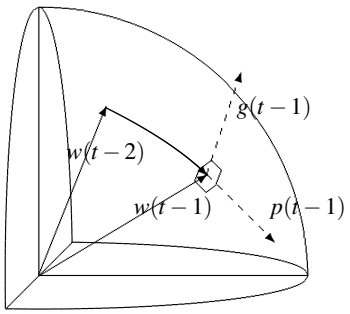

Fig. 1. New search directions at time t

to $w(t-1)$, which prolongs the shortest path leading from $w(t-2)$ to $w(t-1)$, as shown in figure 1. In other words, the new vector $w(t)$ minimizes the function J in the subspace

$$\mathscr{S}(t-1) = \text{span}\{w(t-1), w(t-2), g(t-1)\}. \tag{3}$$

2.2. Conjugate Gradient method

As mentioned above, descent methods usually converge to the global minimum of the Rayleigh quotient. Thus if $\nabla J(t-1) = 0$, the conjugate gradient method has converged. Below, we assume $\nabla J(t-1) \neq 0$. Suppose that the $n \times 3$ matrix $S(t-1) = [w(t-1), p(t-1), g(t-1)]$ is an orthonormal basis of the subspace $\mathscr{S}(t-1)$ defined in equation (3). Below, we show how to compute the vectors $w(t)$, $p(t)$ and $g(t)$, such that the $n \times 3$ matrix $S(t) = [w(t), p(t), g(t)]$ is orthonormal. Since the unitary vector $w(t)$ belongs to $\mathscr{S}(t-1)$, it satisfies

$$w(t) = S(t-1)\,\theta(t), \tag{4}$$

where $\theta(t)$ is an unitary vector of dimension 3. Then substituting equation (4) into equation (1) yields

$$J(S(t-1)\,\theta(t)) = \theta(t)^H C_{ss}^-(t)\,\theta(t), \tag{5}$$

where the 3×3 positive definite matrix $C_{ss}^-(t)$ is defined as

$$C_{ss}^-(t) = S(t-1)^H C_{xs}^-(t) \tag{6}$$

and $C_{xs}^-(t)$ is the $n \times 3$ compressed correlation matrix

$$C_{xs}^-(t) = C_{xx}\,S(t-1). \tag{7}$$

Equation (5) shows that the vector $w(t) = S(t-1)\,\theta(t)$ minimizes the function J if and only if $\theta(t)$ is an unitary eigenvector of the matrix $C_{ss}^-(t)$ associated to the lowest eigenvalue $\lambda(t)$. Thus both $\theta(t)$ and $\lambda(t)$ can be computed by means of any symmetric eigenvalue algorithm (see [6, Chapter 8] for instance).

Let us denote $\{\theta_1(t), \theta_2(t), \theta_3(t)\}$ the coefficients of the vector $\theta(t)$. Since the directions of the vectors $w(t)$ and $w(t-1)$ are different[4], we cannot have both $\theta_2(t) = 0$ and $\theta_3(t) = 0$. Thus the unitary vector $\phi(t)$ introduced below is always well-defined[5]:

$$\phi(t) = \begin{bmatrix} -\sqrt{|\theta_2(t)|^2 + |\theta_3(t)|^2} \\ \theta_1^*(t)\,\frac{\theta_2(t)}{|\theta_2(t)|} \Big/ \sqrt{1 + \left|\frac{\theta_3(t)}{\theta_2(t)}\right|^2} \text{ (or 0 if } \theta_2(t) = 0) \\ \theta_1^*(t)\,\frac{\theta_3(t)}{|\theta_3(t)|} \Big/ \sqrt{1 + \left|\frac{\theta_2(t)}{\theta_3(t)}\right|^2} \text{ (or 0 if } \theta_3(t) = 0) \end{bmatrix}. \tag{8}$$

[4]Indeed, if both vectors had the same direction, then $w(t-1)$ would also minimize the function J in $\mathscr{S}(t-1)$. In this case, $\nabla J(t-1)$ would be orthogonal to $\mathscr{S}(t-1)$. In particular, $\nabla J(t-1)$ would be orthogonal to itself, thus $\nabla J(t-1) = 0$. However we supposed that $\nabla J(t-1) \neq 0$.

[5]The particular form of equation (8) aims at avoiding rounding errors when $\theta_2(t)$ or $\theta_3(t)$ tends to zero.

Then the new descent direction $p(t)$ is defined as follows:

$$p(t) = S(t-1)\phi(t). \tag{9}$$

Indeed, it can be verified that the family $\{w(t), p(t)\}$ is an orthonormal basis of the subspace $\text{span}\{w(t), w(t-1)\}$. Then suppose that $\nabla J(t) \neq 0$ (otherwise the algorithm has converged), and let $g(t) = \nabla J(t) \,/\, \|\nabla J(t)\|$. Since $w(t)$ minimizes J in the subspace $\mathscr{S}(t-1)$, $\nabla J(t) \perp \mathscr{S}(t-1)$. Since both $w(t)$ and $p(t)$ belong to $\mathscr{S}(t-1)$, $g(t)$ is orthogonal to $w(t)$ and $p(t)$. Therefore the matrix

$$S(t) = [w(t), p(t), g(t)] \tag{10}$$

is an orthonormal basis[6]of the subspace $\mathscr{S}(t)$.

2.3. Fast implementation

The dominant cost of the conjugate gradient method as presented above is $3n^2$ flops[7], which is three times the cost of the product of the $n \times n$ matrix C_{xx} by an n-dimensional vector (*cf.* equations (2) and (6)). Note that in some applications such as time series analysis, the correlation matrix C_{xx} presents a particular structure, due to the shift invariance property. In this case, the correlation matrix-vector products are reduced to Toeplitz matrix-vector products, which can be efficiently computed by means of Fast Fourier Transforms (FFT). Thus the dominant cost of our algorithm becomes $O(n \log_2(n))$.

However, to efficiently implement this algorithm, the number of correlation matrix-vector products has to be reduced. The following implementation involves only one such multiplication per iteration. This is achieved by recursively updating 3 auxiliary matrices:

- the $n \times 3$ subspace matrix $S(t)$ defined in equation (10);
- the $n \times 3$ compressed matrix $C_{xs}(t) \triangleq C_{xs}^-(t+1)$:

$$C_{xs}(t) = C_{xx}S(t); \tag{11}$$

- the 3×3 positive definite matrix $C_{ss}(t) \triangleq C_{ss}^-(t+1)$:

$$C_{ss}(t) = S(t)^H C_{xs}(t). \tag{12}$$

These three auxiliary matrices can be efficiently updated. Let

$$\Theta(t) = [\theta(t), \phi(t)]. \tag{13}$$

Substituting equations (4), (9) and (13) into (10) yields

$$S(t) = \left[\, S(t-1)\Theta(t) \,\middle|\, g(t) \,\right]. \tag{14}$$

Then substituting equations (14) and (7) into equation (11) yields

$$C_{xs}(t) = \left[\, C_{xs}^-(t)\Theta(t) \,\middle|\, C_{xx}\,g(t) \,\right]. \tag{15}$$

Once the auxiliary matrices $S(t)$ and $C_{xs}(t)$ have been updated, the gradient direction $\frac{1}{2}\nabla J(t) = C_{xx}w(t) - \lambda(t)w(t)$ is trivially obtained as the first column of the matrix $C_{xs}(t) - \lambda(t)S(t)$.

[6]In practice, as long as the algorithm converges, $\|\nabla J(t)\|$ becomes smaller and smaller. Therefore the normalization of $\nabla J(t)$ introduces rounding errors which make the vector $g(t)$ slowly lose its orthogonality with respect to $w(t)$ and $p(t)$. However this orthogonality can be enforced by means of the following operations:

$$\begin{array}{rcl} g(t) & \leftarrow & \left(I_n - w(t)\,w(t)^H - p(t)\,p(t)^H\right) g(t) \\ g(t) & \leftarrow & g(t) \,/\, \|g(t)\|\,. \end{array}$$

[7]In this paper, a flop is a multiply / accumulate (MAC) operation.

Substituting equations (11), (14), (7) and (6) into (12) yields[8]

$$C_{ss}(t) = \left[\begin{array}{c|c} \Theta(t)^H C_{ss}^-(t)\Theta(t) & \begin{matrix}\times\\ \times\end{matrix} \\ \hline \begin{matrix}\times & \times\end{matrix} & \times \end{array}\right]. \quad (16)$$

The pseudo-code of this fast algorithm is summarized in table 1. Its overall complexity is $n^2 + O(n)$ flops (or $O(n\log_2(n))$ in the case of time series analysis), which can be compared to that of Fuhrmann's conjugate gradient method [7]: $2n^2 + O(n)$.

Table 1. Fast Minor Component Computation

	Eq.	Flops
$[\theta(t), \lambda(t)] = \min\left(\text{eig}\left(C_{ss}^-(t)\right)\right)$		$O(1)$
$\phi(t) = \begin{bmatrix} -\sqrt{\lvert\theta_2(t)\rvert^2 + \lvert\theta_3(t)\rvert^2} \\ \theta_1^*(t)\frac{\theta_2(t)}{\lvert\theta_2(t)\rvert} \Big/ \sqrt{1+\left\lvert\frac{\theta_3(t)}{\theta_2(t)}\right\rvert^2} \\ \theta_1^*(t)\frac{\theta_3(t)}{\lvert\theta_3(t)\rvert} \Big/ \sqrt{1+\left\lvert\frac{\theta_2(t)}{\theta_3(t)}\right\rvert^2} \end{bmatrix}$	(8)	$O(1)$
$\Theta(t) = [\theta(t), \phi(t)]$	(13)	
$S(t)_{(:,1:2)} = S(t-1)\Theta(t)$	(14)	$6n$
$C_{xs}(t)_{(:,1:2)} = C_{xs}^-(t)\Theta(t)$	(15)	$6n$
$C_{ss}(t)_{(1:2,1:2)} = \Theta(t)^H C_{ss}^-(t)\Theta(t)$	(16)	$O(1)$
$\frac{1}{2}\nabla J(t) = C_{xs}(t)_{(:,1)} - \lambda(t) S(t)_{(:,1)}$		n
$S(t)_{(:,3)} = \nabla J(t) \big/ \lVert\nabla J(t)\rVert$		n
$C_{xs}(t)_{(:,3)} = C_{xx}(t) S(t)_{(:,3)}$		n^2
$C_{ss}(t)_{(:,3)} = S(t)^H C_{xs}(t)_{(:,3)}$		$3n$

3. MINOR COMPONENT AND SUBSPACE TRACKING

3.1. Adaptive implementation of the Conjugate Gradient method

In an adaptive context, we consider a sequence of n-dimensional data vectors $\{x(t)\}_{t\in\mathbb{Z}}$, and we are interested in tracking the minor component of its correlation matrix $C_{xx}(t)$. In the literature, this matrix is generally updated according to an exponential windowing:

$$C_{xx}(t) = \beta\, C_{xx}(t-1) + x(t)\,x(t)^H \quad (17)$$

where $0 < \beta < 1$ is the forgetting factor. The principle of the adaptive algorithm described below consists in interlacing the recursive update of the correlation matrix (17) with one step of the conjugate gradient method presented in section 2. Since C_{xx} is now time-varying, we need to distinguish the following auxiliary matrices:

- the $n \times 3$ compressed matrices $C_{xs}^-(t)$ and $C_{xs}(t-1)$:

$$\begin{aligned} C_{xs}^-(t) &= C_{xx}(t)S(t-1) \quad (18)\\ C_{xs}(t-1) &= C_{xx}(t-1)S(t-1) \quad (19)\end{aligned}$$

- the 3×3 positive definite matrices $C_{ss}^-(t)$ and $C_{ss}(t-1)$:

$$\begin{aligned} C_{ss}^-(t) &= S(t-1)^H C_{xx}(t) S(t-1) \quad (20)\\ C_{ss}(t-1) &= S(t-1)^H C_{xx}(t-1) S(t-1). \quad (21)\end{aligned}$$

Substituting equations (17) and (19) into equation (18) yields

$$C_{xs}^-(t) = \beta C_{xs}(t-1) + x(t)\,s(t)^H, \quad (22)$$

where $s(t)$ is the 3-dimensional compressed data vector

$$s(t) = S(t-1)^H x(t). \quad (23)$$

Then substituting equations (17), (21), and (23) into (20) yields

$$C_{ss}^-(t) = \beta C_{ss}(t-1) + s(t)\,s(t)^H. \quad (24)$$

Besides this modification, the remaining of the algorithm presented in section 2 is left unchanged. The pseudo-code of the resulting algorithm is summarized in table 2. Its overall complexity[9] is $n^2 + O(n)$, which is to be compared to that of Chen [8]'s adaptive conjugate gradient method : $2n^2 + O(n)$.

Table 2. Adaptive Minor Component Computation

	Eq.	Flops
$s(t) = S(t-1)^H x(t)$	(23)	$3n$
$C_{xs}^-(t) = \beta\, C_{xs}(t-1) + x(t)\,s(t)^H$	(22)	$3n$
$C_{ss}^-(t) = \beta\, C_{ss}(t-1) + s(t)\,s(t)^H$	(24)	9
Fast Minor Component Computation	Table 1	n^2

3.2. Improving the YAST minor subspace tracker

We are now interested in tracking the r-dimensional minor subspace of the correlation matrix. Recently, we introduced the YAST algorithm as a very fast and precise minor subspace tracker [5]. Below, we show how the adaptive conjugate gradient method presented above can be used to enhance the performance of YAST. The YAST algorithm relies on a principle similar to that introduced in section 2.1: an $n \times r$ orthonormal matrix $W(t)$ spans the r-dimensional minor subspace of $C_{xx}(t)$ if and only if it minimizes the criterion

$$\mathscr{J}(W(t)) = \text{trace}\left(W(t)^H C_{xx}(t) W(t)\right).$$

In particular, the minimum of this criterion is equal to the sum of the r lowest eigenvalues of $C_{xx}(t)$. However, implementing this minimization over all orthonormal matrices is computationally demanding (the complexity is $O(n^2 r)$), and does not lead to a simple recursion between $W(t)$ and $W(t-1)$. In order to reduce the computational cost, this search is limited to the range space of $W(t-1)$ plus one or two additional search directions. It is shown in [5] that this approach leads to a low rank recursion for the subspace weighting matrix, which results in a very low complexity ($O(nr)$ in the case of time series analysis). In [5], the proposed search directions were $x(t)$ and possibly $C_{xx}(t-1)\,x(t)$, which had already been introduced in the case of PSA [13]. However these vectors, whose most energetic part belongs to the signal subspace, prove to be much more suitable for PSA than MSA. In order to enhance the performance of the YAST minor subspace tracker, we propose to replace these vectors by the vector $w(t)$ computed by our adaptive conjugate gradient method, which belongs to the minor subspace. Thus the new subspace tracker consists in interlacing one iteration of the algorithm in table 2 prior to each iteration of the YAST algorithm[10].

4. SIMULATION RESULTS

4.1. Minor Component Computation

In this section, our conjugate gradient algorithm is applied to the computation of the minor eigenvector of a fixed $n \times n$ positive definite matrix C_{xx} (with $n = 25$), randomly chosen. It is compared

[8] In equation (16), the coefficients denoted $\times$ cannot be computed recursively. However, since the matrix $C_{ss}(t)$ is hermitian, only the last column has to be computed, the last row being its conjugate transpose.

[9] This complexity can also be reduced in the case of time series analysis. However the exponential window update (17) does not lead to a simple factorization of the correlation matrix in terms of Toeplitz matrices. A truncated window should be used instead (*cf.* [12, pp. 2]).

[10] Note that the computation of the vector $C_{xx}(t)w(t)$ required by YAST is already included in our conjugate gradient algorithm.

to the constrained steepest descent method presented in section 2.1, and to the conjugate gradient algorithms proposed in [7,8]. For each algorithm the initial vector is chosen randomly, and 10000 iterations are computed. As in [3,4], we calculate the average estimation error

$$\rho(t) = \frac{1}{K}\sum_{k=1}^{K} \frac{\text{trace}\left(W_k(t)^H E_1 E_1^H W_k(t)\right)}{\text{trace}\left(W_k(t)^H E_2 E_2^H W_k(t)\right)},$$

where the number of algorithm runs is $K = 50$, k indicates that the associated variable depends on the particular run, $\|.\|_F$ denotes the Frobenius norm, E_1 (resp. E_2) is the exact[11] $(n-r)$-dimensional principal (resp. r-dimensional minor) subspace basis of C_{xx} (here $r = 1$). Figure 2 shows the evolution of the average errors with respect to the number of iterations. The three conjugate gradient-based algorithms converge faster than the steepest descent method, although Fuhrmann's algorithm [7] seems less stable than Chen's algorithm [8] (the error increases after having reached a minimum). Nevertheless, our conjugate gradient method converges faster than the other algorithms and provides a better accuracy at convergence.

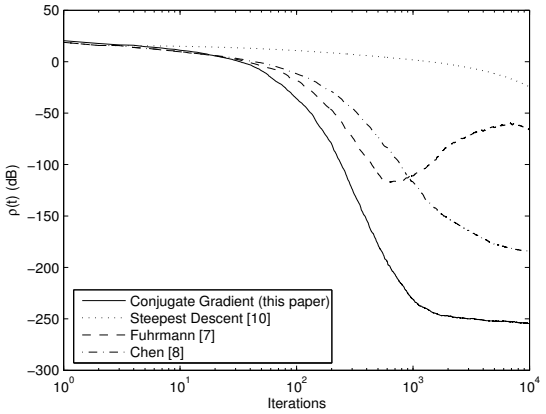

Fig. 2. Minor component Computation

4.2. Subspace Tracking

Below, $x(t)$ is a sequence of $n = 4$ dimensional independent jointly-Gaussian random vectors, centered and with correlation matrix

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

We choose $r = 2$ and $W(0) = \left[I_r, 0_{(r,n-r)}\right]^T$. Four minor subspace trackers are applied to this sequence: QRI [3], NOOja [4], the original, and the modified version of YAST proposed in this paper[12] (called gYAST in figure 3). Again, the experiment is repeated $K = 50$ times. Figure 3 shows the average performance factors. As expected, the estimation error decreases faster in the case of the modified version of YAST. As in [3,4], we also calculated the average departure from orthogonality of the subspace weighting matrix:

$$\eta(t) = \frac{1}{K}\sum_{k=1}^{K} \left\|W_k(t)^H W_k(t) - I_r\right\|_F^2.$$

We observed that the orthonormality error of the four algorithms remains negligible after 10000 iterations (*i.e.* lower than -280 dB).

[11] E_1 and E_2 are obtained by the eigenvalue decomposition (EVD) of C_{xx}.

[12] The QRI algorithm was implemented with parameter $\alpha = 0.99$, NOOja with $\beta = 0.05$ and $\gamma = 0.4$, and YAST with $\beta = 0.99$ and $p = 1$.

5. CONCLUSIONS

In this paper, we introduced a new conjugate gradient method for computing the minor eigenvector of a data correlation matrix. It was shown that this algorithm is less computationally demanding and converges faster than existing methods derived from the conjugate gradient approach. An adaptive version of this algorithm was proposed, which can be used as a pre-processing step for the YAST minor subspace tracker, with a negligible overcost. Our simulations show that the resulting algorithm outperforms other subspace trackers found in the literature, and guarantees the orthonormality of the subspace weighting matrix at each time step.

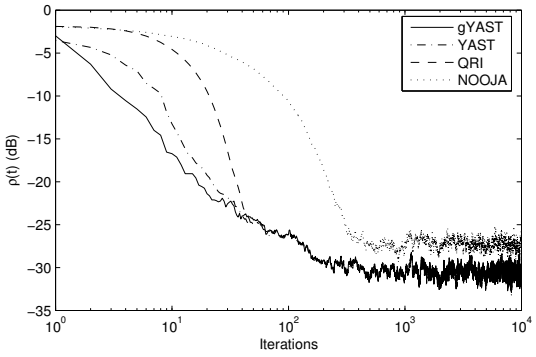

Fig. 3. Subspace Tracking

6. REFERENCES

[1] A-J. Van der Veen, ED. F. Deprettere, and A. L. Swindlehurst, "Subspace based signal analysis using singular value decomposition," *Proc. of IEEE*, vol. 81, no. 9, pp. 1277–1308, Sept. 1993.

[2] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927–935, Nov./Dec. 1992.

[3] P. Strobach, "Square root QR inverse iteration for tracking the minor subspace," *IEEE Trans. Signal Processing*, vol. 48, no. 11, 2000.

[4] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Proc. Letters*, vol. 8, no. 7, pp. 203–206, 2001.

[5] R. Badeau, B. David, and G. Richard, "Yast algorithm for minor subspace tracking," in *Proc. of ICASSP'06*, Toulouse, France, may 2006, vol. III, pp. 552–555, IEEE.

[6] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore and London, third edition, 1996.

[7] D.R. Fuhrmann and B. Liu, "An iterative algorithm for locating the minimal eigenvector of a symmetric matrix," in *Proc. of ICASSP'84*, Dallas, TX, 1984, pp. 45.8.1–4, IEEE.

[8] H. Chen, T. K. Sarkar, S. A. Dianat, and J. D. Brulé, "Adaptive spectral estimation by the conjugate gradient method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 2, pp. 272–284, Apr. 1986.

[9] X. Yang, T. K. Sarkar, and E. Arvas, "A survey of conjugate gradient algorithms for solution of extreme eigen-problems of a matrix," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 10, Oct. 1989.

[10] J. H. Manton, "Optimization Algorithms Exploiting Unitary Constraints," *IEEE Trans. Signal Processing*, vol. 50, no. 3, Mar. 2002.

[11] A. V. Knyazev, "Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method," *SIAM J. Sci. Comput.*, vol. 3, no. 2, pp. 515–541, 2001.

[12] R. Badeau, B. David, and G. Richard, "Fast Approximated Power Iteration Subspace Tracking," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2931–294, Aug. 2005.

[13] R. Badeau, B. David, and G. Richard, "Yet Another Subspace Tracker," in *Proc. of ICASSP'05*, Philadelphia, PA, Mar. 2005, vol. 4, IEEE.