Digital Signal Processing **9,** 297–314 (1999)
Article ID dspr.1999.0348, available online at http://www.idealibrary.com on IDEAL®

A New Look at the Power Method for Fast Subspace Tracking

Yingbo Hua,[*] Yong Xiang,[*] Tianping Chen,[†] Karim Abed-Meraim,[‡] and Yongfeng Miao[§]

[*]Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria 3052, Australia; [†]Department of Mathematics, Fudan University, China; [‡]Telecom, Paris, France; [§]U.S. Wireless Corp., California
E-mail: yhua@ee.mu.oz.au

Hua, Yingbo, Xiang, Yong, Chen, Tianping, Abed-Meraim, Karim, and Miao, Yongfeng, A New Look at the Power Method for Fast Subspace Tracking, *Digital Signal Processing* **9** (1999), 297–314.

A class of fast subspace tracking methods such as the Oja method, the projection approximation subspace tracking (PAST) method, and the novel information criterion (NIC) method can be viewed as power-based methods. Unlike many non-power-based methods such as the Given's rotation based URV updating method and the operator restriction algorithm, the power-based methods with arbitrary initial conditions are convergent to the principal subspace of a vector sequence under a mild assumption. This paper elaborates on a natural version of the power method. The natural power method is shown to have the fastest convergence rate among the power-based methods. Three types of implementations of the natural power method are presented in detail, which require respectively $O(n^2p)$, $O(np^2)$, and $O(np)$ flops of computation at each iteration (update), where n is the dimension of the vector sequence and p is the dimension of the principal subspace. All of the three implementations are shown to be globally convergent under a mild assumption. The $O(np)$ implementation of the natural power method is shown to be superior to the $O(np)$ equivalent of the Oja, PAST, and NIC methods. Like all power-based methods, the natural power method can be easily modified via subspace deflation to track the principal components and, hence, the rank of the principal subspace. ©1999 Academic Press

1. INTRODUCTION

Fast estimation and tracking of the principal subspace (or/and principal components) of a sequence of random vectors is a classic problem which is encountered widely in areas such as data compression, data filtering, parameter estimation, pattern recognition, and neural analysis. In particular, it is important in the area of wireless communications, e.g., [16,17,24].

For this problem, the conventional solutions are available in the 1990 survey paper [1]. The merger of signal processing and neural networks in the early 1990s [2] brought much attention to a method originated by Oja [3] and applied by many others [4]. The Oja method requires only $O(np)$ flops of computation at each iteration (update), where n is the dimension of the vector sequence and p is the dimension of the desired principal subspace. Thorough analyses [5–6] of the Oja method also showed that the Oja method with an arbitrarily small step size (and some mild condition) is globally convergent. However, the convergence speed of its digital implementation, due to the requirement of small step size, is very slow in comparison to many existing numerical methods. If the step size is not small enough, the Oja method diverges.

The Oja method can also be viewed [7,8] as an approximation of gradient descent of a mean square error (MSE) function. Following the gradient descent idea, faster subspace tracking methods have recently been developed, among which are the projection approximation subspace tracking (PAST) method [8] and the novel information criterion (NIC) method [9]. These two methods can both be implemented using only $O(np)$ flops at each iteration. The PAST method was developed from the same MSE function as used for the Oja method, except that a "projection approximation" was applied. The PAST method converges much faster than the Oja method, although the former does not generally converge to an orthogonal basis of the principal subspace. The NIC method is another gradient-based method. The criterion used by the NIC method is unique and different from the MSE function, but it has all the desired properties needed for principal subspace estimation. The NIC method turns out to be a robust generalization of the PAST method. As shown in Appendix B of this paper, the three methods: Oja, PAST, and NIC, can all be viewed as some heuristic variations of a classic method for subspace computation, known as the power method [10]. These methods will be referred to as power-based methods.

It is important to mention that there exist many non-power-based subspace tracking methods, which include the Given's rotation based URV updating method by Stewart [20], the operator restriction algorithm (OPERA) by MacInnes [13], and the maximum likelihood adaptive subspace estimation (MALASE) by Riou and Chonavel [12]. The URV method is now a well-known subspace tracking method, which requires $O(n^2)$ flops at each iteration. The OPERA is one of the latest, which needs $O(np^2)$ flops at each iteration. The MALASE[1] appears to be the fastest non-power-based method, which needs $O(np)$ flops at each iteration. Unlike the power-based methods, however, all of these methods require proper initializations or otherwise they do not converge. Other numerous non-power-based methods can be found in [12–13].

In this paper, we focus on a natural version of the power method and discuss its implementations with varied computational complexities. Like Oja, PAST, and NIC, the natural power method can be implemented with linear complexity of computation, i.e., $O(np)$. Furthermore, the natural power method has the

[1] MALASE is flawed, as discussed later.

fastest convergence rate. In contrast to the non-power-based methods, such as OPERA, the natural power method not only has the minimum computational requirement but it also has a global convergence property.

The concept and implementations of the natural power method are highlighted in this paper and presented in detail in Section 2. A global and exponential convergence property of the natural power method is proven in Appendix A. A link of the power method to the Oja, PAST, and NIC methods is discussed in Appendix B. The natural power method is further modified in Appendix C for principal component tracking. The performance of the natural power method is illustrated in Section 3.

2. THE NATURAL POWER METHOD

The data under consideration are a sequence of $n \times 1$ random vectors: $\{\mathbf{x}(k)\}$. The covariance matrix of the sequence is denoted by $\mathbf{C} = E[\mathbf{x}(k)\mathbf{x}(k)^H]$. The principal subspace spanned by the sequence, of dimension $p < n$, is defined to be the span of the p principal eigenvectors of the covariance matrix. It is known that the first principal eigenvector of the covariance matrix can be obtained by the classic power method as

$$\mathbf{w}(i + 1) = \mathbf{C}\mathbf{w}(i), \tag{1}$$

where the $n \times 1$ weight vector $\mathbf{w}(i)$ is the estimate of the first principal eigenvector at the ith iteration. The norm of the weight vector needs to be scaled during the iterations to prevent it from becoming too large or too small. A natural choice of the scaling can be done as

$$\mathbf{w}(i + 1) = \mathbf{C}\mathbf{w}(i)(\mathbf{w}^H(i)\mathbf{C}^2\mathbf{w}(i))^{-1/2}, \tag{2}$$

where the scaling $(\mathbf{w}^H(i)\mathbf{C}^2\,\mathbf{w}(i))^{-1/2}$ guarantees that $\mathbf{w}(i + 1)$ has norm one. It is known that under a weak condition, the weight vector of (1) converges exponentially to the desired vector (up to a scalar). The convergence rate is governed by the ratio of the first and second largest eigenvalues of the covariance matrix. In tracking applications, one can simply replace the covariance matrix in (2) by its recursive version, i.e., at the $(i + 1)$th iteration,

$$\mathbf{C}(i + 1) = \alpha\mathbf{C}(i) + \mathbf{x}(i + 1)\mathbf{x}(i + 1)^H, \tag{3}$$

where $\mathbf{C}(i)$ is the sample covariance matrix using the data available up to time i, and α is a forgetting factor chosen between zero and one. The algorithm (2) is referred to as the natural power method for computing the first principal component.

For computing a principal subspace of dimension p, one can use a natural generalization of (2) as

$$\mathbf{W}(i + 1) = \mathbf{C}(i + 1)\mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{C}^2(i + 1)\mathbf{W}(i))^{-1/2}, \tag{4}$$

where $\mathbf{W}(i)$ is a $n \times p$ matrix, and the square root inverse of the matrix $\mathbf{W}^H(i)\mathbf{C}^2\mathbf{W}(i)$ is defined by the eigenvalue decomposition of the matrix with its eigenvalues replaced by their square root inverses. It is easy to show that if the initial weight matrix $\mathbf{W}(0)$ satisfies $\mathbf{W}^H(0)\mathbf{C}^2\mathbf{W}(0) > 0$ (positive definite), or equivalently $\mathbf{W}^H(0)\mathbf{C}\mathbf{W}(0) > 0$, then $\mathbf{W}(i)$ remains orthonormal; i.e.,

$$\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{I}, \tag{5}$$

where $i > 0$. The algorithm (4) is referred to as the basic form of the natural power method for subspace tracking.

A global and exponential convergence property of the natural power method is shown in Appendix A. In Appendix B, the power method (1) is shown to be the essence in all of the Oja, PAST, and NIC methods. The rest of this section is focused on digital implementations of the natural power method. These implementations are in great contrast to the analog (differential equations) methods shown in [15]. The analog methods would require analog computers, or otherwise their digital implementations would be too slow, due to the requirement of fine discretization.

2.1. $O(n^2p)$ Implementation—NP1

The direct implementation of (4) requires n^2p flops to compute $\mathbf{C}(i+1)\mathbf{W}(i)$, about $\frac{1}{2}np^2$ additional flops to compute $\mathbf{W}^H(i)\mathbf{C}^2(i+1)\mathbf{W}(i)$ (which is conjugate symmetric), $O(p^3)$ additional flops to compute the inverse square root in (4) (by computing the eigendecomposition and taking the inverse square roots of the eigenvalues), and a further np^2 flops to complete (4). The above approach needs $O(n^2p) + O(np^2) + O(p^3)$ flops.

Alternatively, one can apply the QR factorization of the product $\mathbf{C}(i+1)\mathbf{W}(i)$,

$$\mathbf{C}(i+1)\mathbf{W}(i) = \mathbf{Q}(i+1)\mathbf{R}(i+1), \tag{6}$$

where $\mathbf{Q}(i+1)$ is orthonormal and $\mathbf{R}(i+1)$ is upper (or lower) triangular. Using (6) in (4) yields

$$\mathbf{W}(i+1) = \mathbf{Q}(i+1)\mathbf{H}(i+1), \tag{7}$$

where

$$\mathbf{H}(i+1) = \mathbf{R}(i+1)(\mathbf{R}^H(i+1)\mathbf{R}(i+1))^{-1/2}. \tag{8}$$

It is easy to verify that $\mathbf{H}(i+1)$ is a unitary matrix, and hence, $\mathbf{W}(i+1)$ and $\mathbf{Q}(i+1)$ are equivalent for subspace estimation. Therefore, the direct implementation of the natural power method (4) is equivalent (up to a unitary rotation) to the well-known QR power method [10]:

$$\mathbf{C}(i+1)\mathbf{Q}(i) = \mathbf{Q}(i+1)\mathbf{R}(i+1). \tag{9}$$

This algorithm requires n^2p to compute the product on the left side of (9) and $O(np^2)$ additional flops [1] to compute the QR factorization on the right side.

Each of (4) and (9) guarantees the orthogonality of the weight matrix at each iteration.

2.2. *$O(np^2)$ Implementation—NP2*

To reduce the complexity of the natural power method, we consider (using (3)):

$$\mathbf{C}(i+1)\mathbf{W}(i) = \alpha\mathbf{C}(i)\mathbf{W}(i) + \mathbf{x}(i+1)\mathbf{x}^H(i+1)\mathbf{W}(i). \tag{10}$$

The first term in (10) does not contain any information carried in the new vector $\mathbf{x}(i+1)$ while the second term does. Furthermore, the first term needs many more flops than the second term. This motivates the idea that the first term should be approximated by some past information to save the computational complexity. A simple choice of approximation is

$$\mathbf{C}(i)\mathbf{W}(i) \approx \mathbf{C}(i)\mathbf{W}(i-1) \tag{11}$$

which is clearly valid if the weight matrix $\mathbf{W}(i)$ is slowly varying with i. (Note that the approximation of (11) can be generalized by post-multiplying the right side of (11) by such a matrix $\mathbf{E}(i)$ that minimizes $\|\mathbf{W}(i) - \mathbf{W}(i-1)\,\mathbf{E}(i)\|$. This generalized approximation is used in the bi-iteration method [25]. But for very fast implementations of the natural power method, $\mathbf{E}(i)$ can be replaced by the identify matrix as shown in the sequel of this paper.) With the approximation (11) in (10), the product $\mathbf{C}(i+1)\mathbf{W}(i)$ requires only $3np$ flops to update at each iteration. The approximation (11) also follows from the "projection approximation" $\mathbf{W}^H(i)\mathbf{x}(k) \approx \mathbf{W}^H(k-1)\mathbf{x}(k)$ for all $i \geq k-1$ as proposed in [8].

It is important to note that the approximation (11) is justifiable by the natural power method (4) but not by the QR power method (9). The reasons are as follows.

"Projection approximation" justifiable by the natural power method. Assume that $\mathbf{x}(i)$ is stationary and $\alpha = 1$. One can write that for large i, $(1/(i+1))$ $\mathbf{C}(i+1) \approx \mathbf{U}_1\Sigma_1\mathbf{U}_1^H + \mathbf{U}_2\Sigma_2\mathbf{U}_2^H$, where the first term consists of the principal components, namely, $\mathbf{U}_1$ is the (orthonormal) principal-eigenvector matrix of $\mathbf{C}(i+1)$. According to the discussion in Appendix A for large i,

$$\mathbf{W}(i) \approx \mathbf{U}_1\mathbf{T}(i),$$

where $\mathbf{T}(i)$ is a unitary matrix, and hence,

$$\frac{1}{i+1}\mathbf{C}(i+1)\mathbf{W}(i) \approx \mathbf{U}_1\Sigma_1\mathbf{T}(i)$$

$$(\mathbf{W}^H(i)\mathbf{C}(i+1)^2\mathbf{W}(i))^{-1/2} \approx \frac{1}{i+1}\mathbf{T}(i)^H\Sigma_1^{-1}\mathbf{T}(i)$$

and hence, from (4), $\mathbf{W}(i+1) \approx \mathbf{W}(i)$. The above discussion also shows why the algorithm (17) shown later is asymptotically equivalent to (4).

"Projection approximation" not justifiable by the QR power method. Consider the desired situation $\mathbf{Q}(i) \approx \mathbf{U}_1\mathbf{S}(i)$, where $\mathbf{S}(i)$ is a unitary matrix. Then, for large i, $\mathbf{C}(i+1)\mathbf{Q}(i) = \mathbf{U}_1\Sigma_1\mathbf{S}(i) + \mathbf{P}(i+1)$, where $\mathbf{P}(i+1)$ is small perturbation matrix. By applying the QR factorization (9), we have $\mathbf{Q}(i+1) \approx \mathbf{U}_1\mathbf{S}(i+1)$, where $\mathbf{S}(i+1)$ is a unitary matrix that is rotationally not only affected by the small matrix $\mathbf{P}(i+1)$ but also significantly by Σ_1 unless the latter is proportional (in scalar) to the identity matrix. Therefore, (11) cannot be applied to the QR power method.

To continue the discussion of the natural power method, we now define $\mathbf{Y}(i+1) \triangleq \mathbf{C}(i+1)\mathbf{W}(i)$. Then, with the approximation (11), (10) becomes

$$\mathbf{Y}(i+1) = \alpha\mathbf{Y}(i) + \mathbf{x}(i+1)\mathbf{y}^H(i+1), \tag{12}$$

where

$$\mathbf{y}(i+1) = \mathbf{W}^H(i)\mathbf{x}(i+1). \tag{13}$$

Let $\mathbf{Z}(i+1) \triangleq \mathbf{Y}^H(i+1)\mathbf{Y}(i+1)$. It then follows that

$$\begin{aligned}\mathbf{Z}(i+1) = \alpha^2\mathbf{Z}(i) &+ \mathbf{z}(i+1)\mathbf{y}^H(i+1)\\ &+ \mathbf{y}(i+1)\mathbf{z}^H(i+1) + \gamma(i+1)\mathbf{y}(i+1)\mathbf{y}^H(i+1)\end{aligned} \tag{14}$$

with

$$\mathbf{z}(i+1) = \alpha\mathbf{Y}^H(i)\mathbf{x}(i+1) \tag{15}$$

$$\gamma(i+1) = \mathbf{x}^H(i+1)\mathbf{x}(i+1). \tag{16}$$

Note that like $\mathbf{Y}(i+1)$, the matrix $\mathbf{Z}(i+1)$ also requires only $O(np)$ flops to update at each iteration according to (14). The natural power method (4) now becomes

$$\mathbf{W}(i+1) = \mathbf{Y}(i+1)\mathbf{Z}^{-1/2}(i+1). \tag{17}$$

The above implementation has reduced the complexity to $O(np^2) + O(p^3)$ flops, which is linear in n. Like (4) and (9), (17) also guarantees the orthogonality of the weight matrix at each iteration.

It is important to note that (17) is asymptotically equivalent to (4), provided that $\mathbf{x}(i)$ is wide-sense stationary and α is one.

2.3. O(np) Implementation—NP3

To further reduce the complexity of the natural power method, we need to accommodate asymmetric square root inverse to be applied to (17). We now write

$$\mathbf{Z}^{-1/2}(i+1) = \frac{1}{\alpha}(\mathbf{Z}^{1/2}(i)(\mathbf{I} + \mathbf{b}\mathbf{a}^H + \mathbf{a}\mathbf{b}^H + \gamma\mathbf{a}\mathbf{a}^H)\mathbf{Z}^{1/2^H}(i))^{-1/2}, \tag{18}$$

where the dependence of the vectors on the index $i + 1$ is dropped for convenience, and

$$\mathbf{a} = \frac{1}{\alpha}\mathbf{Z}^{-1/2}(i)\mathbf{y}(i+1) \qquad \mathbf{b} = \frac{1}{\alpha}\mathbf{Z}^{-1/2}(i)\mathbf{z}(i+1). \tag{19}$$

Since $\mathbf{b}\mathbf{a}^H + \mathbf{a}\mathbf{b}^H + \gamma\mathbf{a}\mathbf{a}^H$ is conjugate symmetric and of rank 2, we can write

$$\mathbf{I} + \mathbf{b}\mathbf{a}^H + \mathbf{a}\mathbf{b}^H + \gamma\mathbf{a}\mathbf{a}^H = \mathbf{I} + \lambda_1\mathbf{e}_1\mathbf{e}_1^H + \lambda_2\mathbf{e}_2\mathbf{e}_2^H, \tag{20}$$

where $\mathbf{e}_i$ (orthonormal) and λ_i (real) can be easily computed from the eigendecomposition of a 2×2 matrix as follows. Let $[\mathbf{e}_1\ \mathbf{e}_2] = [\mathbf{a}\ \mathbf{b}]\mathbf{T}$, where $\mathbf{T}$ is a 2×2 matrix. Using this in (20) leads to $\mathbf{RT} = \mathbf{T}\mathrm{diag}(\lambda_1\ \lambda_2)$, where $\mathbf{R}$ is a 2×2 matrix defined by

$$\mathbf{R} = \{[\mathbf{a}\ \mathbf{b}]^H[\mathbf{a}\ \mathbf{b}]\}^{-1}[\mathbf{a}\ \mathbf{b}]^H(\mathbf{b}\mathbf{a}^H + \mathbf{a}\mathbf{b}^H + \gamma\mathbf{a}\mathbf{a})[\mathbf{a}\ \mathbf{b}]$$

and $\mathbf{T}$ is now the eigenvector matrix of $\mathbf{R}$. It is easy to verify that the computational complexity of (20) is negligible in comparison to $O(np)$ when n is large.

With (20), an asymmetric square root inverse of $\mathbf{Z}(i + 1)$ is

$$\mathbf{Z}^{-1/2}(i+1) = \frac{1}{\alpha}[\mathbf{I} - \tau_1\mathbf{e}_1\mathbf{e}_1^H - \tau_2\mathbf{e}_2\mathbf{e}_2^H]\mathbf{Z}^{-1/2}(i), \tag{21}$$

where $\tau_i = 1 - 1/\sqrt{\lambda_i + 1}$. The asymmetric square root inverse is valid in the sense that

$$\mathbf{Z}^{-1/2}(i+1)\mathbf{Z}(i+1)\mathbf{Z}^{-1/2^H}(i+1) = \mathbf{I}. \tag{22}$$

The asymmetric square root inverse of $\mathbf{Z}(i + 1)$ differs from the symmetric square root inverse of $\mathbf{Z}(i + 1)$ by a unitary rotation; i.e., the symmetric square root inverse premultiplied by any unitary matrix still satisfies (22). This unitary rotation tends to weaken the assumption (11). However, unlike the case of the QR power method, this unitary rotation is a slightly rotated version of the identity matrix when $\mathbf{x}(i + 1)\mathbf{x}^H(i + 1)$ is much smaller than $\alpha\mathbf{C}(i)$. If α is close to one and i is large, this rotation is negligible. This is the reason why the algorithm (24) shown later is asymptotically equivalent to (17) and (4).

For analytical convenience, we now use a variation of (17) as

$$\mathbf{W}(i+1) = \mathbf{Y}(i+1)\mathbf{Z}^{-1/2^H}(i+1) \tag{23}$$

which is equivalent to (17) if the symmetric square-root-inverse is used. When an asymmetric square-root-inverse is used, (23) differs from (17) by a postmultiplied unitary rotation.

With the expression (21), we can now write (23) as

$$\begin{aligned}\mathbf{W}(i+1) &= (\alpha\mathbf{Y}(i) + \mathbf{x}\mathbf{y}^H)\mathbf{Z}^{-1/2^H}(i)[\mathbf{I} - \tau_1\mathbf{e}_1\mathbf{e}_1^H - \tau_2\mathbf{e}_2\mathbf{e}_2^H]\,\frac{1}{\alpha}\\ &= \mathbf{W}(i)[\mathbf{I} - \tau_1\mathbf{e}_1\mathbf{e}_1^H - \tau_2\mathbf{e}_2\mathbf{e}_2^H] + \mathbf{x}\mathbf{y}'^H[\mathbf{I} - \tau_1\mathbf{e}_1\mathbf{e}_1^H - \tau_2\mathbf{e}_2\mathbf{e}_2^H]\,\frac{1}{\alpha}, \quad (24)\end{aligned}$$

where the dependence of all vectors on $i + 1$ is dropped for convenience, and

$$\mathbf{y}' = \mathbf{Z}^{-1/2}(i)\mathbf{y}. \quad (25)$$

The expression (24) provides an algorithm which does not involve any matrix–matrix multiplication and in fact requires only $O(np)$ flops of computation at each iteration. Furthermore, if an exact (symmetric or asymmetric) square root inverse of the initial matrix $\mathbf{Z}(0)$ of full rank is used, the algorithm (24) guarantees the orthogonality of the weight matrix at each iteration.

It is important to emphasize that (24) is asymptotically equivalent to (17) and (4), provided that $\mathbf{x}(i)$ is wide-sense stationary and α is equal to one.

3. PERFORMANCE ILLUSTRATION

In this section, we will compare the NP3 of the natural power method with three other power-based methods (Oja, PAST, and NIC) and two non-power-based methods (MALASE and OPERA). The two non-power-based methods have a complexity linear in n. We now assume that the data vector $\mathbf{x}(k)$ satisfies the model

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k),$$

where $\mathbf{s}(k)$ is a $p \times 1$ signal vector and $\mathbf{n}(k)$ is a $n \times 1$ white Gaussian noise vector. We further assume that $p = 2$, $n = 10$, and

$$\mathbf{A} = \begin{bmatrix}\mathbf{R}\\ \mathbf{O}\end{bmatrix}$$

with

$$\mathbf{R} = \begin{bmatrix}\cos\frac{\pi}{6} & \sin\frac{\pi}{6}\\ -\sin\frac{\pi}{6} & \cos\frac{\pi}{6}\end{bmatrix}.$$

The signal vector consists of two independent (randomly selected) Gaussian moving-average processes of order 2. The variances of the two processes are 1.2 and 1, respectively. The element of the noise vector has a variance $\sigma^2 = 0.01$, which corresponds to $SNR \triangleq 10\log_{10}(1/\sigma^2) = 20dB$.[2]

The performance of each method is measured by the subspace error and the orthogonality error that are defined

$$ERROR_{\mathrm{SUB}}(k) = 20\log_{10}\frac{\left\|(\mathbf{I} - \mathbf{W}(k)(\mathbf{W}(k)^H\mathbf{W}(k))^{-1}\mathbf{W}(k)^H)\mathbf{P}_{\mathbf{x}}\right\|_F}{\sqrt{p}}$$

$$ERROR_{\mathrm{ORTH}}(k) = 20\log_{10}\frac{\left\|\mathbf{I} - \mathbf{W}(k)^H\mathbf{W}(k)\right\|_F}{\sqrt{p}},$$

where $\mathbf{P}_{\mathbf{x}}$ is the exact orthogonal projection matrix onto the principal subspace of $\mathbf{x}(k)$. In all the examples shown below, we used $\alpha = 0.99$, $\mathbf{C}(0) = \gamma\mathbf{I}$, where $\gamma = 10$ and $\mathbf{W}(0)$ = random matrix.

Figure 1 compares the performances of the three implementations of the natural power methods: NP1, NP2, and NP3. Figures 1(a)–(c) (the left column) show the subspace tracking errors of NP1, NP2, and NP3 for 50 independent trials. Figure 1(d) shows the averaged tracking error of NP1, NP2, and NP3. Figures 1(e)–(h) (the right column) show the orthogonality errors. As expected, the three versions of the natural power method are all convergent with arbitrary initializations. The NP1 and NP2 guarantee the orthogonality at each iteration. The NP3 yields an orthogonal basis asympototically with arbitrary initializations. Note that, as discussed before, the NP3 also guarantees the orthogonality at each iteration if the initial matrix $\mathbf{Z}(0)$ is properly chosen.

Figure 2 compares the PAST, Oja (step size 0.0001), NIC (step size 0.8), and NP3. In terms of the subspace error, the PAST, NIC, and NP3 have similar convergence rates, and the Oja method has a much slower rate. In terms of the orthogonality error, the NP3 method outperforms all others, especially when the matrix $\mathbf{W}(k)$ is nearly orthogonal.

Figure 3 compares the NP3, OPERA [13], and MALASE [12] (step size 0.01). Due to the arbitrary initializations, OPERA and MALASE fail completely (although simulations in [12,13] show that OPERA and MALASE are locally convergent). It is important to note that as shown in Fig. 3, MALASE in fact yields a constant subspace (easy to prove by using (9) in [12]). MALASE only slightly (not visible from the figure) alters the orthogonality of the basis matrix of the principal subspace. Therefore, despite its $O(np)$ computational efficiency, MALASE is flawed!

As reported in [13], OPERA (in $O(np^2)$) has been tested against many other methods (including PAST). Provided that the initial basis of the principal subspace is "close" to the exact one, OPERA was found to be among the most

[2] This is a reference with respect to one of the two sources. A more meaningful SNR is not necessary.

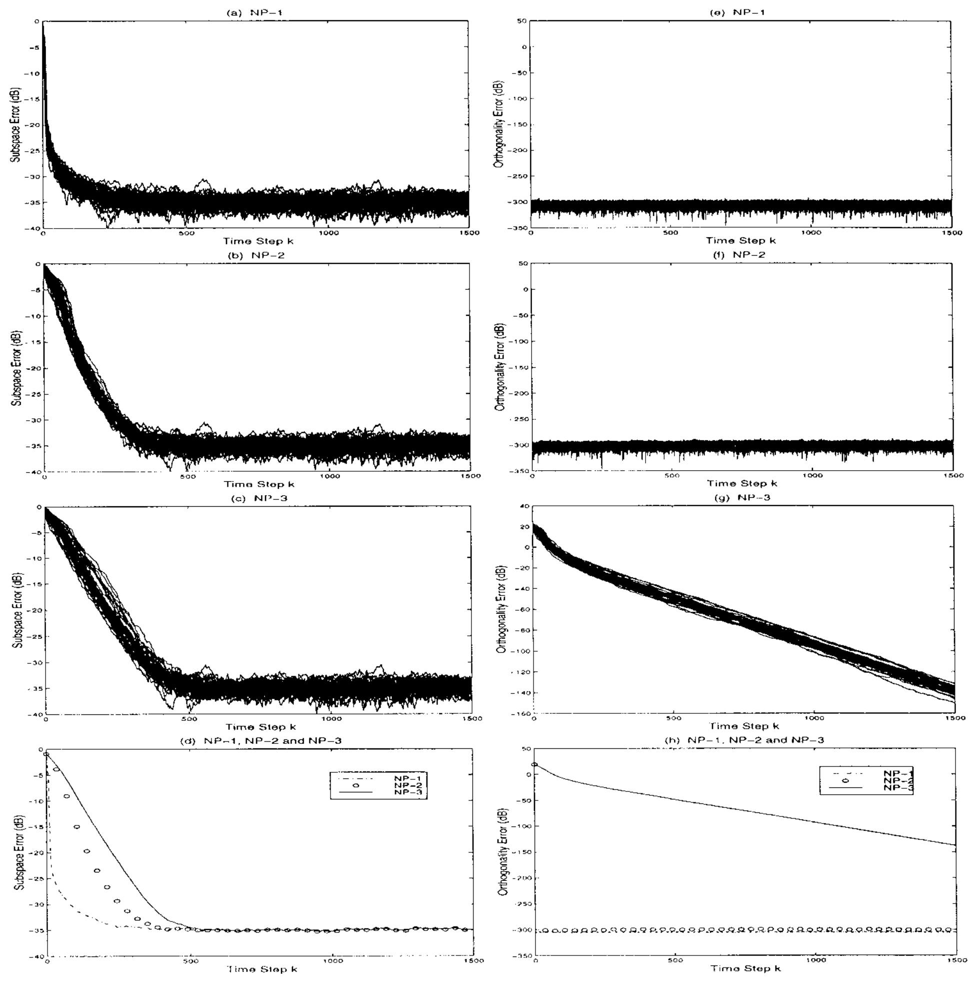

FIG. 1. Performance comparison of NP-1, NP-2, and NP-3.

robust against noise and more robust than PAST—a power-based method. In fact, it is not surprising that OPERA and some other non-power-based methods [23] are "locally" more robust against noise than PAST and other power-based methods. This is because the non-power-based methods are more constrained than the power-based methods, by the old data structure. This is also a reason why the non-power-based methods fail when the initializations are not accurate enough or there is a large sudden change to the subspace structure of the data.

4. FINAL REMARKS

The power method for subspace tracking has been revisited in this paper. Like other power-based methods, the natural power method has a very

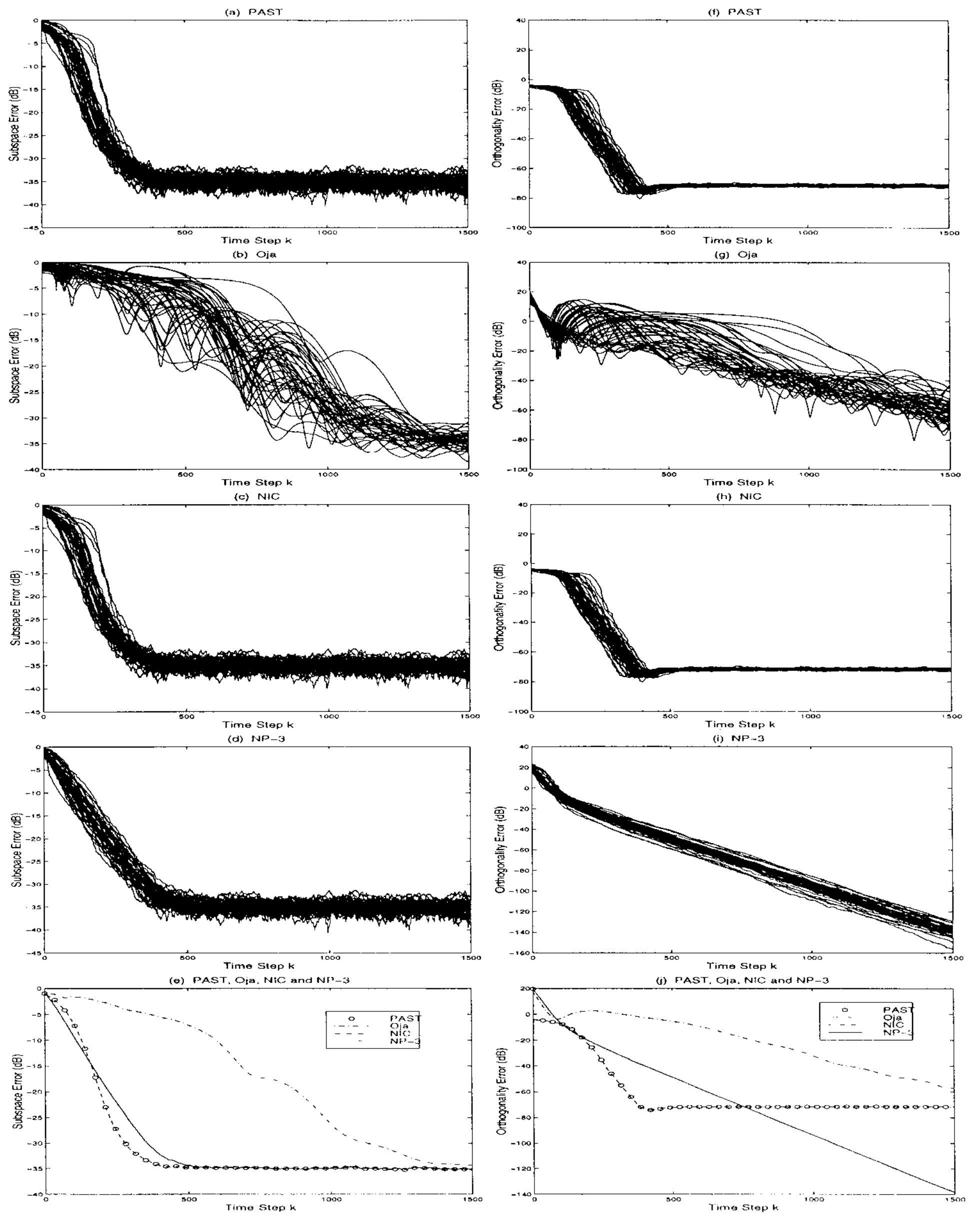

FIG. 2. Performance comparison of PAST, Oja, NIC, and NP-3.

attractive global convergence property important for subspace tracking. The implementation NP3 of the natural power method shown in this paper is as competitive as any fast subspace tracking method as the NP3 only needs $O(np)$ flops at each iteration. Furthermore, the NP3 is guaranteed to be globally convergent which is a property not shared by many non-power-based methods. The natural power method can be easily modified for tracking principal components by using the subspace deflation technique. Such an algorithm is given in Appendix C.

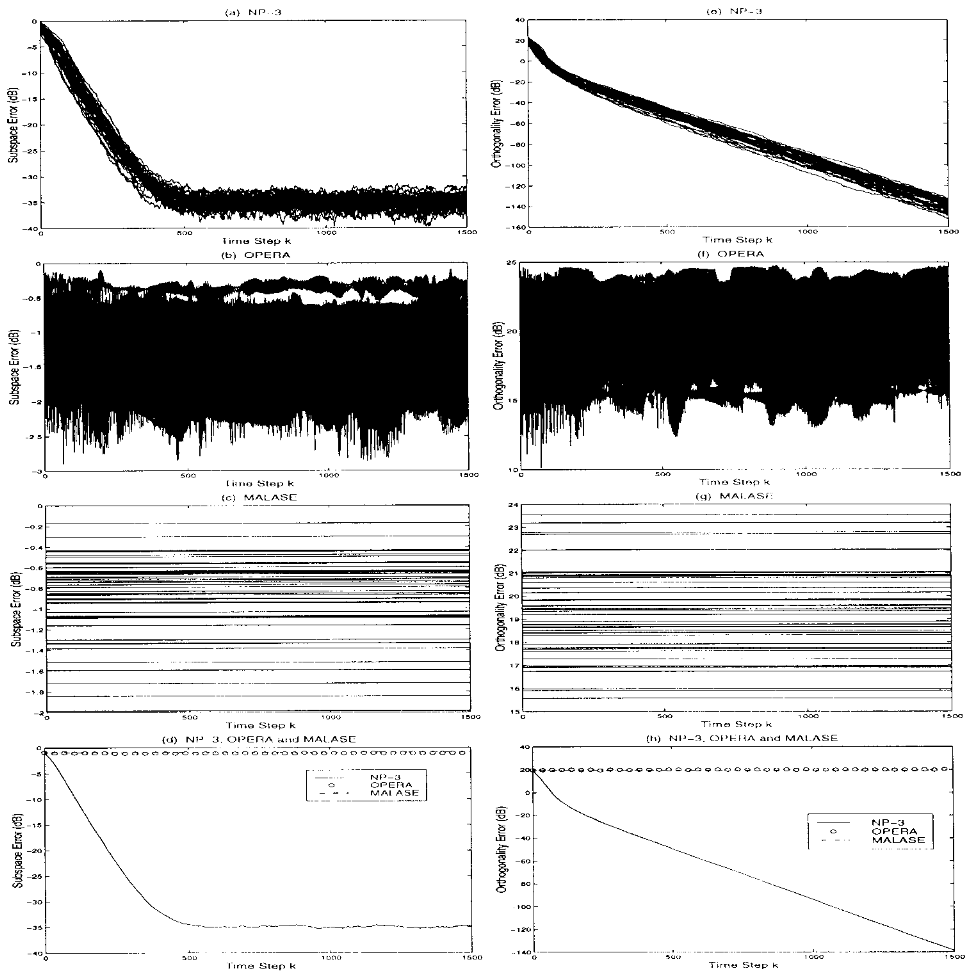

FIG. 3. Performance comparison of NP-3, OPERA, and MALASE.

APPENDIX A: CONVERGENCE PROPERTY

This appendix proves that if (a) the covariance matrix $\mathbf{C}$ remains constant, (b) the pth and $(p + 1)$th eigenvalues of the covariance matrix are distinct, and (c) the initial weight matrix $\mathbf{W}(0)$ meets a mild condition, then the natural power method (4) globally and exponentially converges to the principal subspace. The condition (a) is achievable asymptotically in the tracking context. Assuming that the vector sequence $\mathbf{x}(i)$ is wide sense stationary and the forgetting factor α is one, the recursive equation (3) implies that $\mathbf{C}(i)$ converges to $i\mathbf{C}$, where the factor i has no effect on (4). The condition (b) is satisfied in practice with probability one. The condition (c) will be specified later, which is also satisfied with probability one if the initial matrix is randomly selected.

Denote the eigendecomposition of the covariance matrix $\mathbf{C}$ as

$$\mathbf{C} = [\mathbf{U}_1\ \mathbf{U}_2]\begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}[\mathbf{U}_1\ \mathbf{U}_2]^H, \tag{A1}$$

where the first p principal components are labeled with the subscript "1," and the rest with the subscript "2." Define a new coordinate system for the weight matrix $\mathbf{W}(i)$:

$$[\mathbf{U}_1\ \mathbf{U}_2]^H\mathbf{W}(i) = \begin{bmatrix} \mathbf{U}_1^H\mathbf{W}(i) \\ \mathbf{U}_2^H\mathbf{W}(i) \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1(i) \\ \mathbf{M}_2(i) \end{bmatrix}. \tag{A2}$$

Applying (A2), (4) becomes equivalent to

$$\begin{bmatrix} \mathbf{M}_1(i+1) \\ \mathbf{M}_2(i+2) \end{bmatrix} = \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}\begin{bmatrix} \mathbf{M}_1(i) \\ \mathbf{M}_2(i) \end{bmatrix}\mathbf{G}(i), \tag{A3}$$

where

$$\mathbf{G}(i) = \left(\begin{bmatrix} \mathbf{M}_1(i) \\ \mathbf{M}_2(i) \end{bmatrix}^H \begin{bmatrix} \Sigma_1^2 & \\ & \Sigma_2^2 \end{bmatrix}\begin{bmatrix} \mathbf{M}_1(i) \\ \mathbf{M}_2(i) \end{bmatrix}\right)^{-1/2}.$$

It follows from (5) and (A2) that

$$\mathbf{M}_1(i)^H\mathbf{M}_1(i) + \mathbf{M}_2(i)^H\mathbf{M}_2(i) = \mathbf{I}\,. \tag{A4}$$

What we need to show now is that the $(n - p) \times p$ matrix $\mathbf{M}_2(i)$ converges to zero, which then immediately implies that the $p \times p$ matrix $\mathbf{M}_1(i)$ converges to a unitary matrix and, hence, $\mathbf{W}(i)$ converges to $\mathbf{U}_1$ postmultiplied by a unitary matrix.

Assume that the $p \times p$ matrix $\mathbf{M}_1(0)$ is a nonsingular matrix, which is the condition (c) mentioned previously and can be satisfied with probability one by a randomly chosen $\mathbf{W}(0)$. This implies that $\mathbf{M}_2(0) = \mathbf{T}\mathbf{M}_1(0)$ for some matrix $\mathbf{T}$. Then, from (A3), we can write

$$\begin{aligned} \mathbf{M}_2(1) &= \Sigma_2\mathbf{M}_2(0)\mathbf{G}(0) \\ &= \Sigma_2\mathbf{T}\mathbf{M}_1(0)\mathbf{G}(0) \\ &= \Sigma_2\mathbf{T}\Sigma_1^{-1}\Sigma_1\mathbf{M}_1(0)\mathbf{G}(0) \\ &= \Sigma_2\mathbf{T}\Sigma_1^{-1}\mathbf{M}_1(1) \end{aligned}$$

and, in general,

$$\mathbf{M}_2(i) = \Sigma_2^i\mathbf{T}\Sigma_1^{-i}\mathbf{M}_1(i). \tag{A5}$$

With the assumption that the first p eigenvalues of $\mathbf{C}$ are strictly larger than the others (i.e., the condition (b)), (A5) together with (A4) implies that $\mathbf{M}_2(i)$ converges to zero and is asymptotically in the order of $(\sigma_{p+1}/\sigma_p)^i$, where σ_p and σ_{p+1} ($<\sigma_p$) are the pth and ($p + 1$)th largest eigenvalues of $\mathbf{C}$. We have now established a global and exponential convergence property of the natural power method. This is a great advantage of the natural power method over many other methods.

Based on the convergence proof of the natural power method, it is not difficult to see that the concept of the power method (1) can lead to many forms that are all convergent to the principal subspace. In fact, replacing the matrix $\mathbf{C}^2$ in (4) by $\mathbf{C}$ leads to the algorithm:

$$\mathbf{W}(i + 1) = \mathbf{C}(i + 1)\mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{C}(i + 1)\mathbf{W}(i))^{-1/2}.$$

It can be shown that this algorithm converges to $\mathbf{U}_1\Sigma_1^{1/2}\mathbf{V}$, where $\mathbf{V}$ is a unitary matrix. Furthermore, the square-rooted matrix $(\mathbf{W}^H(i)\mathbf{C}^2\mathbf{W}(i))^{-1/2}$ in (4) can be replaced by any positive definite matrix so long as the norm of the weight matrix $\mathbf{W}(i)$ remains bounded between two positive numbers.

APPENDIX B: OTHER POWER-BASED METHODS

This appendix shows that the power method (1) is the essence of all three methods: Oja, PAST, and NIC.

The Oja method. A heuristic variation of the power method (1) for tracking the first principal component is

$$\mathbf{w}(i + 1) = \mathbf{w}(i) + \eta[\mathbf{C}\mathbf{w}(i) - \mathbf{w}(i)\mathbf{w}^H(i)\mathbf{C}\mathbf{w}(i)]. \tag{B1}$$

The first term on the right side is the previous estimate of the principal eigenvector, which is kept as a memory of the history. The first term in the brackets contains the desired information according to the power method (1). The second term in the brackets is simply the previous estimate with a scalar adjustment $\mathbf{w}^H(i)\mathbf{C}\mathbf{w}(i)$ so that the two terms in the brackets are on a "same" scale. The whole term in the brackets is the desired "new" information. This term is first "properly" scaled by the step size η and then added to the previous estimate to obtain the current estimate. If the previous estimate is already the desired eigenvector with norm one, the "new" information is zero, and hence, no further iteration is necessary. The step size η controls the balance between the history and the new information. With the covariance matrix $\mathbf{C}$ replaced by its instantaneous estimate $\mathbf{x}(i + 1)\mathbf{x}(i + 1)^H$ (corresponding to $\alpha = 0$ in (3)), (B1) is in fact the original Oja method [3].

The Oja method [11] for computing the principal subspace with a predetermined rank p is simply the matrix version of (B1), obtained by replacing the $n \times 1$ weight vector $\mathbf{w}(i)$ by a $n \times p$ weight matrix $\mathbf{W}(i)$; i.e.,

$$\mathbf{W}(i + 1) = \mathbf{W}(i) + \eta[\mathbf{C}\mathbf{W}(i) - \mathbf{W}(i)\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i)]. \tag{B2}$$

With an approximation similar to (11), (B2) can be implemented using only $O(np)$ flops. The Oja method can also be viewed [7–8] as a gradient descent iteration of the MSE function,

$$\begin{aligned} J_{MSE}(\mathbf{W}) &= E[\|\mathbf{x}(k) - \mathbf{W}\mathbf{W}^H\mathbf{x}(k)\|^2] \\ &= tr(\mathbf{C}) - tr(2\mathbf{W}^H\mathbf{C}\mathbf{W} - \mathbf{W}^H\mathbf{C}\mathbf{W}\mathbf{W}^H\mathbf{W}), \quad \text{(B3)} \end{aligned}$$

with the approximation that the weight matrix is orthonormal. It is shown in [5–6] that if the step size is arbitrarily small, the weight matrix of (B2) is globally convergent to an orthonormal basis of the principal subspace. For digital implementation, however, the Oja method converges very slowly due to the fact that only a very small step size is allowed. A larger step size would make the Oja method divergent.

The PAST method. Another heuristic variation of the power method (1) is

$$\mathbf{w}(i+1) = \mathbf{C}\mathbf{w}(i)(\mathbf{w}^H(i)\mathbf{C}\mathbf{w}(i))^{-1}, \quad \text{(B4)}$$

where the inverse factor (similar to the scaling factor used for the Oja method) ensures that the weight vector is "properly" scaled. When the weight vector is the desired eigenvector (up to some scalar), the norm of the weight vector oscillates from one iteration to the next unless the previous weight vector has norm one. Nevertheless, this scaling prevents the norm of the weight vector from growing to infinity or shrinking to zero. For subspace tracking, the weight vector $\mathbf{w}(i)$ is replaced by the weight matrix $\mathbf{W}(i)$; i.e.,

$$\mathbf{W}(i+1) = \mathbf{C}\mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i))^{-1}. \quad \text{(B5)}$$

This is what we call the batch version of the PAST method. It is easy to show that the original PAST method [8] directly follows from (B5) with the approximation (11). The original PAST method is an algorithm of $O(np)$ flops complexity.

The PAST method was originally developed by a gradient descent iteration of the MSE function with the projection approximation. With the same mild conditions as for the natural power method, the PAST method exponentially converges to the principal subspace. But the PAST method does not generally converge to an orthogonal basis of the principal subspace. An example is shown below. With the new coordinate system (A2), (B5) is equivalent to

$$\begin{bmatrix}\mathbf{M}_1(i+1)\\ \mathbf{M}_2(i+1)\end{bmatrix} = \begin{bmatrix}\Sigma_1 & \\ & \Sigma_2\end{bmatrix}\begin{bmatrix}\mathbf{M}_1(i)\\ \mathbf{M}_2(i)\end{bmatrix}\left(\begin{bmatrix}\mathbf{M}_1(i)\\ \mathbf{M}_2(i)\end{bmatrix}^H\begin{bmatrix}\Sigma_1 & \\ & \Sigma_2\end{bmatrix}\begin{bmatrix}\mathbf{M}_1(i)\\ \mathbf{M}_2(i)\end{bmatrix}\right)^{-1}. \quad \text{(B6)}$$

If $\Sigma_2 = 0$, for example, then (B6) leads to

$$\mathbf{M}_1(i+1) = (\mathbf{M}_1(i)^H)^{-1} \qquad \mathbf{M}_2(i+1) = 0 \quad \text{(B7)}$$

which implies that, as i increases, the matrix $\mathbf{M}_1(i)$ oscillates between two matrices unless it is already a unitary matrix.

The NIC method. Following the same idea as shown in (B4) and incorporating some leakage factor (or memory of the history), one obtains

$$\mathbf{w}(i+1) = (1-\beta)\mathbf{w}(i) + \beta\mathbf{C}\mathbf{w}(i)(\mathbf{w}^H(i)\mathbf{C}\mathbf{w}(i))^{-1}, \tag{B8}$$

where the leakage factor β is between (0, 1). This leakage prevents the oscillation associated with the PAST method from happening (a proof of which is available in [14]). The matrix version of (B8), i.e.

$$\mathbf{W}(i+1) = (1-\beta)\mathbf{W}(i) + \beta\mathbf{C}\mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{C}\mathbf{W}(i))^{-1}, \tag{B9}$$

is simply the batch version of the NIC method presented in [9]. With approximation (11), the NIC method can be easily implemented using only $O(np)$ flops at each iteration [9].

It is worth mentioning that the NIC method was originally based on the gradient ascend iteration of the criterion (NIC):

$$J_{NIC}(\mathbf{W}) = tr[\log(\mathbf{W}^H\mathbf{C}\mathbf{W})] - tr(\mathbf{W}^H\mathbf{W}). \tag{B10}$$

The properties of this criterion are discussed at length in [9]. In particular, this criterion has only one local maximum which is also global. At the global maximum, the weight matrix is orthonormal and spans the principal subspace. The leakage factor β is the same as the step size required for scaling the gradient of the NIC. The NIC method is shown in [9] to be globally convergent to an orthonormal basis of the principal subspace when the step size is arbitrarily small. While simulations have suggested that the NIC method converges to the desired basis for all values of β chosen between (0, 1), a rigorous proof of this property is still under investigation [14].

APPENDIX C: TRACKING THE PRINCIPAL COMPONENTS

This section derives an algorithm for tracking the principal components of a sequence of random vectors by using the natural power method. Denote by $\mathbf{w}_m(k)$ the estimate of the mth principal eigenvector at time k. Following the concept of the natural power method (2) and the subspace deflation technique [4], we can write

$$\overline{\mathbf{w}}_m(k+1) = \mathbf{C}_m(k+1)\mathbf{w}_m(k) \tag{C.1}$$

$$\mathbf{w}_m(k+1) = \overline{\mathbf{w}}_m(k+1)/\left\|\overline{\mathbf{w}}_m(k+1)\right\|, \tag{C.2}$$

where

$$\mathbf{C}_m(k+1) = \alpha\mathbf{C}_m(k) + \mathbf{x}(k+1)\mathbf{x}(k+1)^H(\mathbf{I} - \mathbf{W}_m(k)\mathbf{W}_m(k)^H) \tag{C.3}$$

$$\mathbf{W}_m(k) = [\mathbf{w}_1(k) \cdots \mathbf{w}_{m-1}(k)]. \tag{C.4}$$

Applying the approximation $\overline{\mathbf{w}}_m(k) \cong \mathbf{C}_m(k)\mathbf{w}_m(k)$, (C.1) becomes

$$\overline{\mathbf{w}}_m(k+1) = \alpha\overline{\mathbf{w}}_m(k) + \mathbf{x}(k+1)(g_m(k+1) - \mathbf{y}_m(k+1)^H\mathbf{c}_m(k), \tag{C.4}$$

where

$$g_m(k+1) = \mathbf{x}(k+1)^H\mathbf{w}_m(k) \tag{C.5}$$

$$\mathbf{y}_m(k+1) = \mathbf{W}_m(k)^H\mathbf{x}(k+1) \tag{C.6}$$

$$\mathbf{c}_m(k) = \mathbf{W}_m(k)^H\mathbf{w}_m(k). \tag{C.7}$$

Equations (C.2) and (C.4)–(C.7) should be run successively for all desired components ($m = 1, 2, \ldots$) at each iteration k.

The eigenvalues (up to a common scalar) of the covariance matrix $\mathbf{C}$ of the sequence $\mathbf{x}(k)$ can be updated as follows. From (C.2), one can write

$$\lambda_m(k+1) = \mathbf{w}_m(k)^H\mathbf{C}_m(k+1)\mathbf{w}_m(k) = \mathbf{w}_m(k)^H\overline{\mathbf{w}}_m(k+1). \tag{C.8}$$

Using (C.4) in (C.8), applying the approximation $\lambda_m(k) \cong \mathbf{w}_m(k)^H\overline{\mathbf{w}}_m(k)$ and assuming that $\mathbf{c}_m(k) = \mathbf{W}_m(k)^H\,\mathbf{w}_m(k)$ is small, one can replace (C.8) by

$$\lambda_m(k+1) = \alpha\lambda_m(k) + |g_m(k+1)|^2. \tag{C.9}$$

It is easy to prove that under a mild condition and $\alpha = 1$, the above algorithm (for large k) exponentially converges to the eigenvectors and eigenvalues of $k\mathbf{C}$. Not surprisingly, our simulations confirmed that this algorithm converges much faster than the APEX method [4] that is based on the Oja method (B.1). Similar to what is shown in [21], the above method can be used with any eigenvalue-based detection methods [22] for tracking the rank of the principal subspace. The above method has a similar performance as PASTd [21].

ACKNOWLEDGMENT

The authors acknowledge a useful discussion with Dr. C. MacInnes and a matlab program of OPERA received from him.

REFERENCES

1. Comon, P., and Golub, G. H. Tracking a few extreme singular values and vectors in signal processing. *Proc. IEEE* **78,** No. 8, (1990), 1327–1343.
2. Juang, B. H., Kung, S. Y., and Kamm, C. A. (Eds.), *Proc. 1991 IEEE Workshop on Neural Networks for Signal Processing, Princeton, NJ, Sept. 30–Oct. 2, 1991.*
3. Oja, E. A simplified neuron model as a principal component analyzer, *J. Math. Biol.* **15** (1982), 267–273.
4. Kung, S. Y., Diamantaras, K. I., and Taur, J. S. Adaptive principal component extraction (APEX) and applications. *IEEE Trans. Signal Process.* **42,** No. 5 (1994), 1202–1217.
5. Yan, W. Y., Helmke, U., and Moore, J. B. Global analysis of Oja's flow for neural networks. *IEEE Trans. Neural Networks* **5,** No. 5 (1994), 674–683.

6. Chen, T., Hua, Y., and Yan, W. Global convergence of Oja's subspace algorithm for principal component extraction. *IEEE Trans. Neural Networks* **9,** No. 1 (1998), 58–67.
7. Xu, L. Least mean square error reconstruction principle for self-organizing neural nets. *Neural Networks* **6** (1993), 627–648.
8. Yang, B. Projection approximation subspace tracking. *IEEE Trans. Signal Process.* **43,** No. 1 (1995), 95–107.
9. Miao, Y. and Hua, Y. Fast subspace tracking and neural network learning by a novel information criterion. *IEEE Trans. Signal Process.* **46,** No. 7 (1998), 1967–1979.
10. Golub, G. H. and Van Loan, C. F. *Matrix Computations,* Johns Hopkins Univ. Press, Baltimore, MD, (1989).
11. Oja, E. Neural networks, principal components, and subspaces. *Int. J. Neural Systems* **1,** No. 1 (1989), 61–68.
12. Riou, C. and Chonavel, T. Fast adaptive eigenvalue decomposition: A maximum likelihood approach, in *Proc. of IEEE ICASSP'97,* pp. 3565–3568.
13. MacInnes, C. S. Fast, accurate subspace tracking using operator restriction analysis, *Proc. of IEEE ICASSP'98,* pp. 1357–1360.
14. Chen, T. and Hua, Y. Convergence analysis of the NIC algorithm for subspace tracking, manuscript.
15. Helmke, U. and Moore, J. B. *Optimization and Dynamical Systems,* Springer-Verlag, New York/Berlin, (1994).
16. Wang, X., and Poor, H. V. Blind adaptive multiuser detection in multipath CDMA channels based on subspace tracking. *IEEE Trans. Signal Process.* **46,** No. 11 (1998), 3030–3044.
17. Liu, H. and Xu, G. A subspace method for signal waveform estimation in synchronous CDMA systems. *IEEE Trans. Commun.* **44,** No. 10 (1996), 1346–1354.
18. Champagne, B. Adaptive eigendecomposition of data covariance matrices based on first-order perturbations. *IEEE Trans. Signal Process.* **42** (1994), 2758–2770.
19. DeGroat, R. D. Noniterative subspace tracking. *IEEE Trans. Signal Process.* **40** (1992), 571–577.
20. Stewart, G. W. An updating algorithm for subspace tracking. *IEEE Trans. Signal Process.* **40** (1992), 1535–1541.
21. Yang, B. An extension of the PASTd algorithm to both rank and subspace tracking. *IEEE Signal Process. Lett.* **2,** No. 9 (1995), 179–182.
22. Wax, M. and Kailath, T. Detection of signals by information theoretic criteria. *IEEE Trans. Signal Process.* **33** (1985), 387–392.
23. Tufts, D., Real, E., and Cooley, J. Fast approximate subspace tracking (FAST), in *Proc of the ICASSP'97, April 1997,* pp. 547–550.
24. Hua, Y., Abed-Meraim, K., and Wax, M. Blind system identification using minimum noise subspace. *IEEE Trans. Signal Process.* **45,** No. 3 (1997), 770–773.
25. Strobach, P. Bi-iteration SVD subspace tracking algorithms. *IEEE Trans. Signal Process.* **45,** No. 5 (1997), 1222–1240.