THE FAST DATA PROJECTION METHOD FOR STABLE SUBSPACE TRACKING

Xenofon G. Doukopoulos

France Telecom R&D
4 rue du Clos Courtel - BP 91226
Cesson-Sévigné, France
xenofon.doukopoulos@rd.francetelecom.com

George V. Moustakides

Dept. of Computer & Communication Engineering
University of Thessaly,
37 Glavani & 28 October str, 38221 Volos, Greece
moustaki@uth.gr

ABSTRACT

In this article we consider the Data Projection Method (DPM), which constitutes a simple and reliable means for adaptively estimating and tracking subspaces. Specifically we propose a fast and numerically robust implementation of DPM. Existing schemes can track subspaces corresponding either to the largest or the smallest singular values. DPM, on the other hand, with a simple change of sign in its step size, can switch from one subspace type to the other. Our fast implementation of DPM preserves the simple structure of the original DPM having also a considerably lower computational complexity. The proposed version provides orthonormal vector estimates of the subspace basis which are *numerically stable*. In other words, our scheme does not accumulate roundoff errors and therefore preserves orthonormality in its estimates. In fact, our scheme constitutes the *only* numerically stable, low complexity, algorithm for tracking subspaces corresponding to the smallest singular values. In the case of tracking subspaces corresponding to the largest singular values, our scheme exhibits the fastest convergence-towards-orthonormality among all other subspace tracking algorithms of similar complexity.

1. INTRODUCTION

1.1 Problem definition

In a typical application of subspace-based adaptive signal processing, we are receiving, sequentially, observation vectors $\mathbf{y}(n) \in \mathbb{R}^N$ with covariance matrix $\mathbf{R} = \mathbb{E}\{\mathbf{y}(n)\mathbf{y}^t(n)\}$ and singular value decomposition of the form (SVD)

$$\mathbf{R} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_n \end{bmatrix} [\mathbf{U}_s \quad \mathbf{U}_n]^t, \qquad (1)$$

with $\mathbf{\Lambda}_s = \text{diag}\{\ _1, \ldots, \ _{L_s}\}$, $\mathbf{\Lambda}_n = \text{diag}\{\ _{L_s+1}, \ldots, \ _N\}$, where $\ _1 \geq \ _2 \geq \cdots \geq \ _N \geq 0$ are the singular values of $\mathbf{R}$. The diagonal matrices $\mathbf{\Lambda}_s, \mathbf{\Lambda}_n$ therefore contain the L_s largest and the $L_n = N - L_s$ smallest singular values of $\mathbf{R}$ respectively; while $\mathbf{U}_s, \mathbf{U}_n$ contain the corresponding singular vectors. The matrices $\mathbf{U}_s, \mathbf{U}_n$ are both orthonormal, constituting *orthonormal* bases for the corresponding subspaces. The problem we would like to solve is now the following:

> *Assuming that $\mathbf{y}(n)$ is available sequentially, we would like to provide adaptive estimates either for $\mathbf{U}_s$ or $\mathbf{U}_n$.*

Perhaps the most common case encountered in practice corresponds to the following data model

$$\mathbf{y}(n) = \mathbf{x}(n) + \mathbf{w}(n), \qquad (2)$$

where $\mathbf{x}(n)$ is a sequence of vectors of length N lying in an L_s-dimensional linear subspace and $\mathbf{w}(n)$ are i.i.d. white noise vectors with independent elements. In this case the SVD in (1) takes the special form

$$\mathbf{R} = [\mathbf{U}_s \quad \mathbf{U}_n] \begin{bmatrix} \mathbf{D}_s + \ ^2\mathbf{I}_{L_s} & \mathbf{0} \\ \mathbf{0} & \ ^2\mathbf{I}_{L_n} \end{bmatrix} [\mathbf{U}_s \quad \mathbf{U}_n]^t, \qquad (3)$$

where $\mathbf{I}_K$ denotes the identity matrix of size K; $\mathbb{E}\{\mathbf{x}(n)\mathbf{x}^t(n)\} = \mathbf{U}_s\mathbf{D}_s\mathbf{U}_s^t$ is the SVD of the covariance matrix of $\mathbf{x}(n)$; $\ ^2$ the noise power and $L_n = N - L_s$. Matrix $\mathbf{U}_s$ is then said to span the *signal* subspace whereas $\mathbf{U}_n$ the *noise* subspace.

From now on, with a slight abuse of notation, we will call the subspace $\mathbf{U}_s$ corresponding to the largest singular values "the signal subspace" and the subspace $\mathbf{U}_n$ corresponding to the smallest singular values "the noise subspace", keeping of course in mind that these names are correct only in the case of the model defined in (2).

1.2 Literature review

In order to avoid the excessively high computational complexity $O(N^3)$ needed by the direct SVD, alternative schemes requiring less operations were developed. If L denotes the rank of the subspace we are interested in, algorithms requiring a wide variety of complexities have already been proposed in the literature. Since, usually, $L \ll N$, we classify the schemes requiring $O(N^2L)$ or $O(N^2)$ operations as *high complexity*; algorithms with complexity $O(NL^2)$ as *medium complexity*; and finally methods with $O(NL)$ operations as *low complexity* (the latter schemes are also known in the literature as *fast subspace tracking algorithms*). Due to space limitation, we mainly emphasize here the low complexity $O(NL)$ class. An exhaustive literature review addressing also the other two classes can be found in [1, Ch. 2], while [2] constitutes an excellent review of the literature up to 1990.

The great majority of articles addressing the problem of subspace tracking focus mainly on signal subspace, while the literature intended for the noise subspace is unfortunately very limited. Starting with the former, let us first introduce two methods coming from the *medium* complexity $O(NL^2)$ class. The Data Projection Method (DPM) [3] will serve as the basis for the novel fast algorithmic scheme we are going to develop and will be introduced in detail in the next section. The most popular algorithm of the medium complexity class was proposed by Karasalo [4]. Karasalo's algorithm offers the best performance to cost ratio [2] and thus serves as a point of reference for all subsequent low complexity $O(NL)$ techniques. Its overall complexity is $O(NL^2 + L^3)$, with the L^3 part coming from the need to perform an SVD on an $(L+1) \times (L+2)$ matrix.

Low complexity subspace tracking schemes are clearly very attractive since they can lend themselves to real-time processing. The Projection Approximation Subspace Tracking (PAST) algorithm is a well known approach for signal subspace tracking proposed in [5]. The main advantage of this $O(NL)$ scheme is its simple structure having a single parameter to be specified. However the estimates offered by this method are not orthonormal. The next two algorithms of interest are MALASE [6] and PROTEUS-2 [7]. Both algorithms have a rather complicated structure with the former having four parameters to be specified and the latter just one. The two algorithms provide orthonormal estimates for the subspace basis. The final algorithm is the Low Rank Adaptive Filter (LORAF) proposed in [8]. This algorithm has also a very difficult structure and exhibits a very sensitive behavior as far as orthonormality of the estimates is concerned.

When we refer to noise subspace tracking, low complexity $O(NL)$ schemes are either unstable (they diverge) [9] (modification of PAST); or numerically nonrobust [10, 11, 12], that is, they lose orthonormality due to round-off error accumulation, a fact that eventually leads to divergence. From the latter three cases, the FRANS algorithm proposed in [12] has an overall best performance which, in the case of noise subspace tracking, as was said, is still numerically unstable.

2. FAST DPM

2.1 Adaptive orthogonal iteration

If $\mathbf{R}$ is a non-negative definite matrix and we are interested in computing a subspace basis $\mathbf{U}_L$ of rank L corresponding either to the signal or to the noise subspace, then there exists a very simple iterative scheme from Numerical Analysis that can perform this task. The method, known as *orthogonal iteration* [13], has the following variant that is suitable for our problem of interest:

$$\mathbf{U}_L(n) = \text{orthonormalize}\{(\mathbf{I} \pm \ \mathbf{R})\mathbf{U}_L(n-1)\}, \quad (4)$$

where $\ > 0$ is a "small" scalar parameter (step size); $\mathbf{U}_L(n)$ is the estimate of the subspace basis at the n-th iteration; finally the "$+$" sign generates estimates for the signal subspace whereas the "$-$" for the noise subspace.

It can be shown that convergence of this iterative scheme is *exponential* at a rate which is approximately equal to $-\ (\ _L - \ _{L+1})$ in the case of the signal subspace and $-\ (\ _{N-L} - \ _{N-L+1})$ in the case of the noise subspace [13].

When matrix $\mathbf{R}$ is unknown and, instead, we have the observed data vector sequence $\mathbf{y}(n)$, we can replace $\mathbf{R}$ in (4) with an adaptive estimate $\mathbf{R}(n)$ that satisfies $\mathbb{E}\{\mathbf{R}(n)\} = \mathbf{R}$. This leads to the *adaptive* orthogonal iteration algorithm

$$\mathbf{U}_L(n) = \text{orthonormalize}\{(\mathbf{I} \pm \ \mathbf{R}(n))\mathbf{U}_L(n-1)\}. \quad (5)$$

Clearly depending on the choice of $\mathbf{R}(n)$ we can obtain different subspace tracking algorithms.

2.2 The DPM and the Fast DPM algorithm

The simplest selection for $\mathbf{R}(n)$ is the *instantaneous* estimate of the covariance matrix, that is, $\mathbf{R}(n) = \mathbf{y}(n)\mathbf{y}^t(n)$ which gives rise to the DPM algorithm [5]

$$\mathbf{U}_L(n) = \text{orthonormalize}\{(\mathbf{I} \pm \ \mathbf{y}(n)\mathbf{y}^t(n))\mathbf{U}_L(n-1)\}, \quad (6)$$

where orthonormalization is performed using Gram-Schmidt. Due to this latter requirement the overall computational complexity of DPM becomes $O(NL^2)$.

Our main contribution consists in offering an alternative orthonormalization process, which reduces the complexity to the desired $O(NL)$ level. The algorithm we propose is the following:

$$\begin{aligned}
\mathbf{r}(n) &= \mathbf{U}_L^t(n-1)\mathbf{y}(n) && (7)\\
\mathbf{T}(n) &= \mathbf{U}_L(n-1) \pm \frac{\bar{}}{\|\mathbf{y}(n)\|^2}\mathbf{y}(n)\mathbf{r}^t(n) && (8)\\
\mathbf{a}(n) &= \mathbf{r}(n) - \|\mathbf{r}(n)\|\mathbf{e}_1 && (9)\\
\mathbf{Z}(n) &= \mathbf{T}(n) - \frac{2}{\|\mathbf{a}(n)\|^2}[\mathbf{T}(n)\mathbf{a}(n)]\mathbf{a}^t(n) && (10)\\
\mathbf{U}_L(n) &= \text{normalize}\{\mathbf{Z}(n)\}, && (11)
\end{aligned}$$

where $\mathbf{e}_1 = [1\,0 \cdots 0]^t$. The first two equations correspond to the part $(\mathbf{I} \pm \ \mathbf{y}(n)\mathbf{y}^t(n))\mathbf{U}_L(n-1)$ of the original DPM with the only difference being the use of a normalized step size $\ = \bar{}/\|\mathbf{y}(n)\|^2$; Relations (9)-(11) replace the orthonormalization process of the original algorithm. We recognize in (10) the use of the Householder transformation $\mathbf{I} - \frac{2}{\|\mathbf{a}(n)\|^2}\mathbf{a}(n)\mathbf{a}^t(n)$ which is crucial for the orthonormalization process. Finally, with the term "normalize" we simply mean the *normalization* of each column of the matrix $\mathbf{Z}(n)$; operation that requires only $O(NL)$ computations. This reduces the overall complexity to $O(NL)$, thus gaining an order of magnitude as compared to the original DPM algorithm.

2.3 Numerical stability

Our Fast DPM (FDPM) version, depicted in (7)-(11), as is the case with all subspace tracking algorithms belonging to the low complexity class $O(NL)$, provides orthonormal estimates if it is initialized with an orthonormal matrix. If for some reason orthonormality is lost, or the adaptation is initialized with a matrix that is not orthonormal, then the algorithm *converges* towards an orthonormal matrix. This should be compared to the original DPM algorithm where orthonormality is assured at every time step due to Gram-Schmidt. Regarding convergence towards orthonormality we have the following important theorem:

Theorem 1 *The FDPM algorithm, for sufficiently small step size , has an exponential convergence rate towards orthonormality which is of the form $c_1 + \ c_2 + o(\ )$, with c_1, c_2 independent of and $c_1 < 0$.*

Proof: The proof can be found in [14]. ■

As a consequence of Theorem 1 we have that the orthonormality error power

$$e_o(n) = \mathbb{E}\{\|\mathbf{U}_L^t(n)\mathbf{U}_L(n) - \mathbf{I}_L\|_{\mathrm{F}}^2\}, \quad (12)$$

where $\|\cdot\|_{\mathrm{F}}$ stands for the Frobenius norm, in FDPM tends to zero as $O(e^{(c_1 + \ c_2)n})$. Due to the fact that c_1 is *strictly* negative we have a guaranteed convergence towards zero, even if the step size is zero. In other words the convergence rate towards orthonormality, for our FDPM scheme, is practically insensitive to changes in the step size (for small).

As we are going to see in the simulations section, the previous property *does not hold* for all other subspace tracking schemes of similar complexity. In particular, all low complexity $O(NL)$ algorithms, as well as Karasalo's scheme (of complexity $O(NL^2)$), have a convergence rate towards orthonormality which is of the form $\ c\ +o(\ )$ with $c<0$. This suggests that any drastic change in the step size of the corresponding algorithm induces an equivalent drastic change in the convergence speed towards orthonormality.

2.4 Behavior of the estimates $\mathbf{U}_L(n)$

The fact that FDPM does not produce exactly orthonormal estimates (unless it is initialized with such a matrix) is not very crucial. The reason is that orthonormality is reached well before the estimates $\mathbf{U}_L(n)$ converge to the exact subspace basis $\mathbf{U}_L$. We have the following lemma regarding this point:

Lemma 1 *Consider the DPM adaptation described in (6). Then its mean trajectory satisfies the following iteration*

$$\mathbb{E}\{\mathbf{U}_L(n)\} \approx \text{orthonormalize}\{(\mathbf{I}\pm\ \ \mathbf{R})\mathbb{E}\{\mathbf{U}_L(n-1)\}\}. \quad (13)$$

Proof: The proof makes use of the Independence Assumption; details can be found in [14]. ∎

Lemma 1 suggests that the *mean* estimates $\mathbb{E}\{\mathbf{U}_L(n)\}$ of DPM (and therefore FDPM) satisfy the orthogonal iteration defined in (4) thus, according to the discussion in Subsection 2.1, we conclude that $\mathbf{U}_L(n)$ converges *in the mean* to the exact $\mathbf{U}_L$ at an exponential rate of the form of $\ c\ +o(\ )$, where $c<0$. Comparing this rate to the one we had for orthonormality (i.e. $c_1\ +\ \ c_2$), we conclude that for sufficiently small step size $\ $ we have $c_1\ +\ \ c_2 \ll\ c$, suggesting that FDPM converges to an orthonormal matrix well before the algorithm converges to its steady state.

In adaptive algorithms another quantity that is of primal importance is the *steady state estimation error power*. The estimation error power at time n is defined as

$$e_p(n)=\mathbb{E}\{\|\mathbf{U}_L(n)\mathbf{U}_L^t(n)-\mathbf{U}_L\mathbf{U}_L^t\|_F^2\}. \quad (14)$$

In other words, we measure the error power between the subspace *projection operators* $\mathbf{U}_L(n)\mathbf{U}_L^t(n)$ and $\mathbf{U}_L\mathbf{U}_L^t$. It is quite fortunate that we can find a closed form expression for the limiting value of this quantity.

Theorem 2 *Consider the DPM adaptation defined in (6) and let* $\ _1,\dots,\ _N$ *be the singular values of the data covariance matrix* $\mathbf{R}=\mathbb{E}\{\mathbf{y}(n)\mathbf{y}^t(n)\}$, *then in the case of estimating the signal subspace of dimension L (using the "+" sign) we have that*

$$\lim_{n\to} \mathbb{E}\{\|\mathbf{U}_L(n)\mathbf{U}_L^t(n)-\mathbf{U}_L\mathbf{U}_L^t\|_F^2\} = \ _{i=1}^{L}\ _{j=L+1}^{N} \frac{\ _i\ _j}{\ _i-\ _j}, \quad (15)$$

whereas in the case of the noise subspace of dimension L (using the "−" sign) it becomes

$$\lim_{n\to} \mathbb{E}\{\|\mathbf{U}_L(n)\mathbf{U}_L^t(n)-\mathbf{U}_L\mathbf{U}_L^t\|_F^2\} = \ _{i=1}^{N-L}\ _{j=N-L+1}^{N} \frac{\ _i\ _j}{\ _i-\ _j}. \quad (16)$$

Proof: The proof can be found in [14]. ∎

3. SIMULATION RESULTS

In this section we present several simulations in order to verify the validity of our theoretical developments of the previous sections. We consider the signal plus noise model of (2) with $N=8$ where the random signal $\mathbf{x}(n)$ lies on an $L=4$ dimensional linear subspace. We assume that the singular values of the signal subspace are $\mathbf{D}_s=\text{diag}\{250,180,120,60\}$ and that the noise is white, Gaussian, with variance $\ ^2$. All subsequent graphs are the result of averaging 100 independent runs.

3.1 Signal subspace tracking

We compare FDPM against the following $O(NL)$ schemes: PAST, PROTEUS-2, MALASE, LORAF-3; and also the more computationally demanding version of Karasalo of complexity $O(NL^2)$. Fig. 1 depicts a typical case of the relative performance of the competing schemes. We plot the projection estimation error power as defined in (14). The noise variance is selected $\ ^2=30$ and at iteration 1500 we apply an abrupt change to the exact matrix $\mathbf{U}_s$, preserving its orthonormal structure, in order to observe the tracking capabilities of all algorithms. FDPM exhibits an overall better performance than its $O(NL)$ rivals, following at the same time very closely Karasalo's $O(NL^2)$ method.

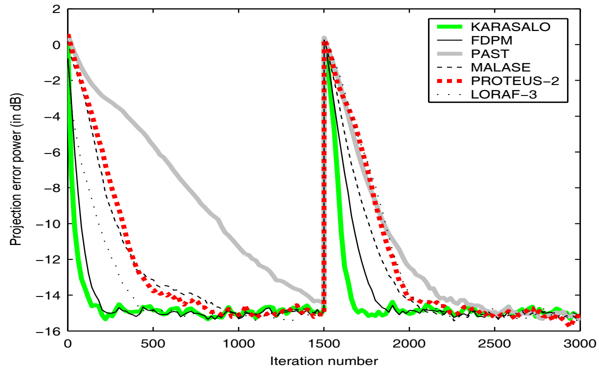

Figure 1: Performance of the signal subspace tracking schemes.

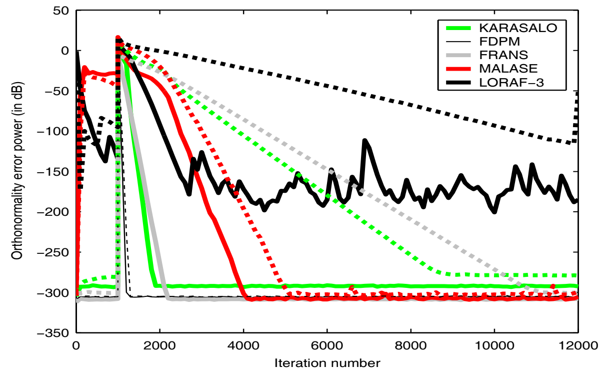

Figure 2: Deviation from orthonormality of the signal subspace tracking schemes.

In Fig. 2 we plot the orthonormality error power, as defined in (12), for FDPM, FRANS [12], MALASE, LORAF-3 and Karasalo's algorithm (PAST does not provide orthonormal estimates whereas PROTEUS-2 has an *extremely* slow convergence). At iteration $n = 1000$ we change abruptly $\mathbf{U}_L(1000)$ into a non-orthonormal matrix in order to observe the convergence towards orthonormality. We run each algorithm with two drastically different values of its corresponding step size, namely (solid line) and $\frac{}{10}$ (dashed line). We can see that, except LORAF-3, all algorithms practically attain orthonormality within machine accuracy level. FDPM is by far the fastest converging. The interesting point however is that the convergence speed of FDPM changes only slightly with the drastic change in its step size. All other algorithms exhibit a significantly reduced convergence speed when the smaller step size is employed, thus corroborating our conclusions presented in Subsection 2.3.

3.2 Noise subspace tracking

The algorithms we compare here are the DPM (requiring $O(NL^2)$ operations) the FRANS algorithm [12] and finally our FDPM version. The signal model is exactly as in the previous subsection therefore the noise subspace is of rank $L = 4$ and has a multiple singular value equal to ^{2} which we select equal to 1. In order to demonstrate the numerical stability of FDPM and the numerical instability of FRANS, we perform simulations similar to the last example of the previous subsection. Namely, at iteration 1500 we replace the estimate $\mathbf{U}_L(1500)$ with a non-orthonormal matrix to examine again convergence towards orthonormality.

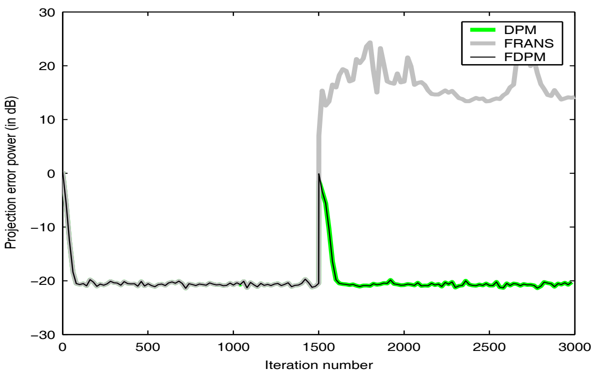

Figure 3: Performance of the noise subspace tracking schemes.

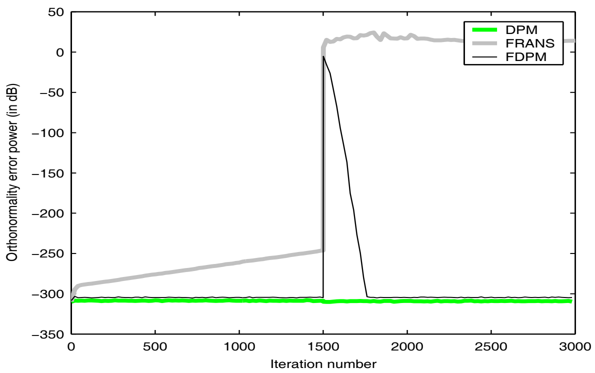

Figure 4: Deviation from orthonormality of the noise subspace tracking schemes.

Fig. 3 depicts the projection estimation error power, while Fig. 4 the corresponding orthonormality error power for exactly the same simulation. We can see that FDPM follows closely DPM and very quickly recovers orthonormality (DPM orthonormalizes at every step). FRANS, on the other hand, gradually loses orthonormality by accumulating round-off errors, as we can verify by the increasing line in Fig. 4 until time 1500. If at some point FRANS has a non-orthonormal estimate then it becomes completely unstable, generating meaningless subspace estimates. This is evident from Fig. 3 the part after time 1500.

4. CONCLUSION

In this work, we have considered the problem of adaptive subspace tracking. Our contribution consists in developing a fast, numerically stable orthonormalization technique for the DPM algorithm [3] reducing its overall complexity from $O(NL^2)$ to $O(NL)$. For noise subspace tracking our scheme is the *only* $O(NL)$ complexity algorithm that is numerically stable achieving an orthonormality error within machine accuracy. Our algorithmic scheme has a very simple structure and exhibits the fastest convergence-towards-orthonormality speed among all existing $O(NL)$ subspace tracking algorithms.

5. ACKNOWLEDGEMENT

The authors are indebted to Prof. Thierry Chonavel for allowing the use of his Matlab code thus facilitating, considerably, the simulation task.

REFERENCES

[1] X.G. Doukopoulos, "Power Techniques for Blind Channel Estimation in Wireless Communication Systems," PhD thesis, IRISA-INRIA, University of Rennes 1, France, 2004.

[2] P. Comon and G.H. Golub. "Tracking a Few Extreme Singular Values and Vectors in Signal Processing," *Proceedings of IEEE*, vol. 78, Aug. 1990.

[3] J.F. Yang and M. Kaveh, "Adaptive Eigensubspace Algorithms for Direction or Frequency Estimation and Tracking," *IEEE Trans. ASSP*, vol. 36, pp. 241-251, Feb. 1988.

[4] I. Karasalo, "Estimating the Covariance Matrix by Signal Subspace Averaging", *IEEE Trans. ASSP*, vol. 34, pp. 8-12, Feb. 1986.

[5] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. Sig. Proc.*, vol. 43, pp. 95-107, Jan. 1995.

[6] T. Chonavel, B. Champagne, and C. Riou, "Fast Adaptive Eigenvalue Decomposition: A Maximum Likelihood Approach," *Signal Processing*, vol. 83, pp. 307-324, 2003.

[7] B. Champagne and Q.G. Liu, "Plane Rotation-Based EVD Updating Schemes for Efficient Subspace Tracking," *IEEE Trans. Sig. Proc.*, vol. 46, pp. 1886-1900, July 1998.

[8] P. Strobach, "Low-Rank Adaptive Filters," *IEEE Trans. Sig. Proc.*, vol. 44, pp. 2932-2947, Dec. 1996.

[9] E. Oja, "Principal Components, Minor Components, and Linear Neural Networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.

[10] K. Abed-Meraim, S. Attallah, A.Chkeif, and Y. Hua, "Orthogonal Oja Algorithm," *IEEE Sig. Proc. Let.*, vol. 7, pp. 116-119, May 2000.

[11] S. Attallah and K. Abed-Meraim, "Fast Algorithms for Subspace Tracking," *IEEE Sig. Proc. Let.*, vol. 8, pp. 203-206, July 2001.

[12] S. Attallah, K. Abed-Meraim, "Low Cost Adaptive Algorithm for Noise Subspace Estimation," *Electronics Letters*, vol. 38, pp. 609-611, June 2002.

[13] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd edn, The John Hopkins University Press, 1996.

[14] X.G. Doukopoulos and George V. Moustakides, "Fast and Stable Subspace Tracking," sumbitted to the *IEEE Trans. Sig. Proc.*, under review.