# Improved $\psi$–APEX Algorithm for Digital Image Compression

Simone Fiori, Saverio Costa, and Pietro Burrascano *

Dept. of Industrial Engineering – University of Perugia (Italy)
E-mail: simone@eealab.unian.it

## Abstract

*In this work we derive an improvement of $\psi$–APEX principal component analysis neural algorithms [8], based on a laterally-connected neural architecture, which arises from an optimization theory specialized for this topology. Such a class contains, as a special case, an APEX-like algorithm, but it also contains a subclass of algorithms that show interesting convergence features when compared with the original one.*

## 1. Introduction

Data reduction techniques, as the Karhunen-Loéve Transform (KLT), aim at providing an efficient representation of the data. The classical approach for evaluating the KLT requires the computation of the input data covariance matrix and then the application of a numerical procedure to extract the eigenvalues and the corresponding eigenvectors; reduction is obtained by projecting the data on the only eigenvectors associated with the most significant eigenvalues. When large data sets are handled, this approach is not practicable because the dimensions of the covariance matrix become too large to be manipulated.

In order to overcome these problems, neural-network-based approaches were proposed. Neural Principal Component Analysis (PCA) is a second-order adaptive statistical data processing technique introduced by Oja [12] that helps to remove the second-order correlation among given random processes. In fact, consider the stationary multivariate random process $\mathbf{x}(t) \in \mathcal{R}^p$ and suppose its covariance matrix $\mathbf{R} = E_{\mathbf{x}}[(\mathbf{x} - E_{\mathbf{x}}[\mathbf{x}])(\mathbf{x} - E_{\mathbf{x}}[\mathbf{x}])^T]$ exists bounded. If $\mathbf{R}$ is not diagonal, then the components of $\mathbf{x}(t)$ are statistically correlated. This second-order redundancy may be partially (or completely) removed by computing a linear operator $\mathbf{Q}$ such that the new random signal defined by $\mathbf{y}(t) \stackrel{\text{def}}{=} \mathbf{Q}^T(\mathbf{x}(t) - E_{\mathbf{x}}[\mathbf{x}]) \in \mathcal{R}^m$ has uncorrelated components, with $m \leq p$ arbitrarily selected [1, 2, 3, 4, 9, 10, 12, 14, 15]. The data-stream can then be reconstructed by the simple synthesis formula $\hat{\mathbf{x}}(t) = \mathbf{Q}\mathbf{y}(t) + E_{\mathbf{x}}[\mathbf{x}]$.

Among others, Kung and Diamantaras proposed in [11] a principal component analyzer, implemented by means of a laterally connected linear neural network having the topology shown in Figure 1, termed Adaptive Principal component EXtractor (APEX). In [5] a generalization of the standard APEX algorithm has been presented, but to the best of our knowledge special attention has not been paid to particularization nor tests have been performed in order to discover their features.

Starting from their work, in [8] we justified by means of an optimization principle the APEX learning algorithm and extended the family of these algorithms yielding a class of PCA learning rules that we called $\psi$–APEX. Here we aim to further extend this class and to show the improvements achieved. Through computer simulations we also compare APEX and new $\psi$–APEX class by performing adaptive digital image compression.

## 2. Principal Component Analysis and the Kung-Diamantaras' Learning Algorithm

On the basis of Rubner-Tavan's laterally-connected PCA topology [13], Kung and Diamantaras developed the principal component neural network described by the following input-output relationships (see Figure 1):

$$\mathbf{z}(t) = \mathbf{W}^T(t)\mathbf{x}(t) \text{ and } \mathbf{y}(t) = \mathbf{z}(t) + \mathbf{L}^T(t)\mathbf{y}(t) \ , \tag{1}$$

with a proper unsupervised learning rule. The zero-mean input vector $\mathbf{x}(t) \in \mathcal{R}^p$, the output vector $\mathbf{y}(t) \in \mathcal{R}^m$ (with $m \leq p$), the direct-connection $p \times m$ weight-matrix $\mathbf{W}(t)$ and the lateral-connection $m \times m$ strictly upper-triangular weight-matrix $\mathbf{L}(t)$ are evaluated at the same temporal instant $t$. The columns of $\mathbf{W}$ and $\mathbf{L}$ are named

*This research was supported by the Italian MURST.

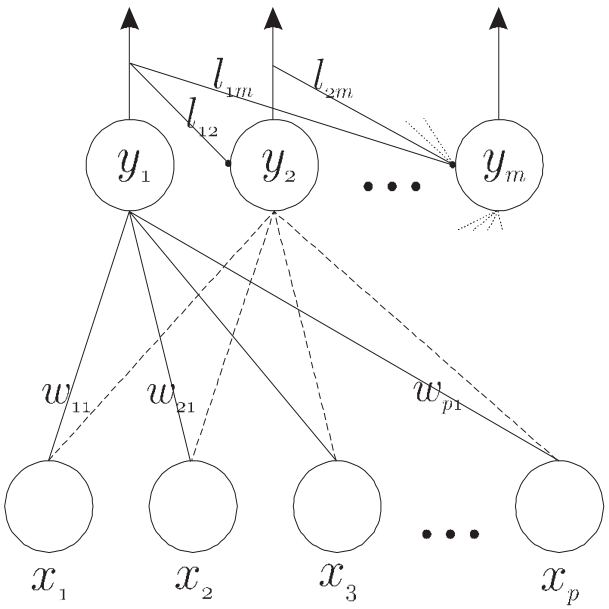


**Figure 1. The laterally-connected topology.**

in the following way: $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_m]$, $\mathbf{L} = [\mathbf{0}\ \mathbf{l}_2\ \cdots\ \mathbf{l}_m]$. The original learning rules [5] for the weight-matrices $\mathbf{W}$ and $\mathbf{L}$ are:

$$\Delta\mathbf{W} = \eta[\mathbf{X}\tilde{\mathbf{Y}} - \mathbf{W}\tilde{\mathbf{Y}}^2]\ ,\ \Delta\mathbf{L} = -\eta\mathtt{SUT}[\mathbf{Y}\tilde{\mathbf{Y}}] - \eta\mathbf{L}\tilde{\mathbf{Y}}^2\ , \tag{2}$$

where $\eta$ is a positive learning rate, $\mathbf{X}$ is a $p \times m$ matrix, $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are $m \times m$ matrices defined by:

$$\mathbf{X} = [\underbrace{\mathbf{x}\ \mathbf{x}\ \cdots\ \mathbf{x}}_{m}]\ ,\ \mathbf{Y} = [\underbrace{\mathbf{y}\ \mathbf{y}\ \cdots\ \mathbf{y}}_{m}]\ ,\ \tilde{\mathbf{Y}} = \mathrm{diag}(y_1, y_2, \ldots, y_m)\ ,$$

and $\mathtt{SUT}[\cdot]$ returns the strictly upper-triangular part of the matrix contained within.

Kung and Diamantaras were able to prove the convergence of the above algorithm under some conditions [5, Theorem 4.3]; in particular, we can shortly restate their results saying that: *If the rate $\eta$ is chosen so small that the behavior of the algorithm is asymptotically stable, the initial entries of $\mathbf{W}$ are small random numbers and $\mathbf{L}(0) = \mathbf{0}$, then in the mean sense:*

$$\lim_{t\to\infty}\mathbf{L}(t) = \mathbf{0}\ ,\ \lim_{t\to\infty}\mathbf{W}(t) = \mathbf{Q}\ ,\ \lim_{t\to\infty}E_{\mathbf{x}}[\mathbf{y}(t)\mathbf{y}^T(t)] = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)\ ,$$

where $\mathbf{R}\mathbf{q}_k = \lambda_k\mathbf{q}_k$, with $\mathbf{R}$ being the data covariance matrix and $\lambda_k$ being its largest eigenvalues. These results are here referred to as KDR.

In [11], Kung and Diamantaras also proposed a sequential version of the APEX algorithm, described by the following relationships, which hold for the $k^{\mathrm{th}}$ neuron:

$$\tilde{\mathbf{y}}_k(t) = \tilde{\mathbf{W}}_k^T\mathbf{x}(t)\ ,\ y_k(t) = \mathbf{w}_k^T(t)\mathbf{x}(t) - \mathbf{l}_k^T(t)\tilde{\mathbf{y}}_k(t) \tag{3}$$

where $\tilde{\mathbf{y}}_k$ is the vector of the previously extracted $k-1$ principal components, and $\tilde{\mathbf{W}}_k$ is the weight-matrix connecting the inputs to the first $k-1$ outputs and does not include $\mathbf{w}_k$; only $\mathbf{w}_k$ and $\mathbf{l}_k$ are trained when the $k^{\mathrm{th}}$ neuron is interested by learning. The $t^{\mathrm{th}}$ iteration of the learning algorithm is described by:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta_k(y_k(t)\mathbf{x}(t) - y_k^2(t)\mathbf{w}_k(t))\ , \tag{4}$$

$$\mathbf{l}_k(t+1) = \mathbf{l}_k(t) + \eta_k(y_k(t)\tilde{\mathbf{y}}_k(t) - y_k^2(t)\mathbf{l}_k(t)) \tag{5}$$

Moreover, in [11] Kung and Diamantaras presented a version of the APEX algorithm endowed with a variable learning rate defined by:

$$\eta_k(t+1) = \frac{\eta_k(t)}{\gamma + y_k(t)^2\eta_k(t)} \tag{6}$$

with $\gamma$ set to a constant value smaller of, but close to, 1 (for instance 0.99). As we shall see in the section devoted to experimental results, the use of an adaptive learning stepsize for the APEX algorithm dramatically improves its performances, at the expense of a major computational complexity.

## 3. The Improved $\psi$–APEX Class

In the following we briefly recall the $\psi$–APEX algorithms family and explain the proposed improvement.

A PCA transformation is such that the transformed signals $\mathbf{z} = \mathbf{W}^T\mathbf{x}$ are characterized by maximum variance. Furthermore, from definitions, it is known that any PCA vector $\mathbf{w}_k$ must be orthogonal with respect to each other and should exhibit unitary norm. [12]. These can be thought of as separate objectives to be achieved by means of the laterally-connected neural network.

An optimization principle for describing the problem at hand can be properly defined by examining the structure of a generic output signal $y_k$ from equation (1). Squaring and taking the expected value yields:

$$E_{\mathbf{x}}[y_k^2] = E_{\mathbf{x}}[(\mathbf{w}_k^T\mathbf{x})^2] + E_{\mathbf{x}}[(\mathbf{l}_k^T\mathbf{y})^2] + 2E_{\mathbf{x}}[(\mathbf{w}_k^T\mathbf{x})(\mathbf{l}_k^T\mathbf{y})] \ . \tag{7}$$

The first term at the right-hand side contains the power of the transformed signal $z_k = \mathbf{w}_k^T\mathbf{x}$, while the second term contains a linear combination of the cross-correlation of the outputs, in fact it holds that $E_{\mathbf{x}}[(\mathbf{l}_k^T\mathbf{y})^2] = \mathbf{l}_k^T E_{\mathbf{x}}[\mathbf{y}\mathbf{y}^T]\mathbf{l}_k$. By definition of PCA, the first term has to be maximized under the constraint $\mathbf{w}_k^T\mathbf{w}_k = 1$ [10], while the second one must be zeroed.

### 3.1 The $\psi$–APEX learning rules

In contrast to the heuristic derivation of Kung-Diamantaras, in [8] we presented a learning algorithm based on criterion optimization. In fact, we proposed to adapt the *direct-connection* weight-matrix $\mathbf{W}$, in order to only maximize the transformed data variance, by means of the following learning rule:

$$\mathbf{W}(t+1) = \mu\mathbf{X}(t)\tilde{\mathbf{Y}}(t) + \mathbf{W}(t)[\mathbf{I}_m - \mu\tilde{\mathbf{W}}(t)\tilde{\mathbf{Z}}(t)] \ , \ \tilde{\mathbf{Z}} \stackrel{\text{def}}{=} \text{diag}(z_1, z_2, \ldots, z_m) \ , \tag{8}$$

and to adapt the *lateral-connection* weight-matrix $\mathbf{H}$, in order to only make the components of the response vector $\mathbf{y}$ be uncorrelated, according to the following rule:

$$\mathbf{L}(t+1) = -\mu\texttt{SUT}[\mathbf{Y}(t)\tilde{\mathbf{Y}}(t)] + \mathbf{L}(t)[\mathbf{I}_m - \mu\tilde{\mathbf{\Psi}}(t)] \ , \tilde{\mathbf{\Psi}} \stackrel{\text{def}}{=} \text{diag}(\psi_1, \psi_2, \ldots, \psi_m) \ , \tag{9}$$

where the $\psi_k$ are arbitrary functions that at least guarantee the stability of the network [8] and $\mu$ is a positive learning rate. $\mathbf{I}_m$ represents an $m \times m$ identity matrix.

A learning rule with the structure (8)-(9) with a generic $\psi$ function was termed $\psi$–APEX. While overcoming the analysis ability of APEX algorithm, the networks belonging to $\psi$–APEX class were not able to correctly extract more than 4-5 components in a reasonable number of iterations, due to the inherent weakness of the decorrelating action performed by the lateral connections.

### 3.2. Improved $\psi$–APEX learning algorithms

Here we propose to adapt the *direct-connection* weight-matrix $\mathbf{W}$ to maximize the powers of the transformed signal by maximizing the following objective function:

$$J(\mathbf{W}, \mathbf{L}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} E_{\mathbf{x}}[y_k^2] + \sum_{k=1}^{m}(1 - \mathbf{w}_k^T\mathbf{w}_k)\mu_k \ , \tag{10}$$

with respect to $\mathbf{W}$ only. Note that the $\mu_k$ are Lagrange multipliers whose optimal values can be easily found ([8] and references therein). This leads to same learning equation (8). Then, we choose to adapt the *lateral-connection* weight-matrix $\mathbf{L}$ only, in order to *minimize* a cost function defined as:

$$C(\mathbf{W}, \mathbf{L}) \stackrel{\text{def}}{=} \rho\sum_{k=1}^{m} E_{\mathbf{x}}[y_k^2] + \sum_{k=1}^{m}(\mathbf{l}_k^T\mathbf{l}_k)E_{\mathbf{x}}[\psi_k] \ , \tag{11}$$

where a set of $m$ Lagrange multipliers $E_{\mathbf{x}}[\psi_k]$ has been introduced for the constraints $\|\mathbf{l}_k\|^2 = 0$, to preserve the KDR. The constant $\rho$ has been introduced because there is no reason to weight the terms $E_{\mathbf{x}}[y_k^2]$ in the same way in (11) as in (10), and this constitutes the major improvement on the earliest version which we discuss in this paper. As will be shown by computer simulations, this improvement allows to obtain better performances at no additional computational complexity cost.

As can be seen by recalling standard Kuhn-Tucker theory, there are no theoretical reasons to force the $\psi_k$ functions to assume any particular value. This observation tells that the choice of a set of functions $\psi_k$ doesn't

change the set of the stationary points of the learning system. Conversely, it is clear that it can heavily affect the learning dynamics, leading to a family of adaptive algorithms that can exhibit a variety of interesting behaviors.

The objective function $C$ can be minimized, with respect to the variable matrix $\mathbf{L}$, by means of a gradient steepest descent method; this way, the following new learning rule arises:

$$\Delta\mathbf{L} = -\eta\rho\mathtt{SUT}[\mathbf{Y}\tilde{\mathbf{Y}}] - \eta\mathbf{L}\tilde{\mathbf{\Psi}} \ , \ \tilde{\mathbf{\Psi}} = \mathrm{diag}(\psi_1, \psi_2, \ldots, \psi_m) \ . \tag{12}$$

As for APEX algorithms, a sequential version for the improved $\psi$–APEX rules can be easily derived.

### 3.3. Convergence analysis and discussion on the choice of $\psi_k's$

In this Section the convergence properties of the proposed class of neural PCA algorithms is addressed. The mean sense convergence of the learning equations to the expected solution is proven by the following Theorem.

**Theorem 1** *Let $\mathbf{R} \stackrel{\mathrm{def}}{=} E_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T]$ be the covariance matrix of the stationary zero-mean random signal $\mathbf{x}(t)$ in (1), with eigenpairs $(\mathbf{q}_1, \lambda_1)$, $(\mathbf{q}_2, \lambda_2)$, $\ldots$, $(\mathbf{q}_p, \lambda_p)$ ordered so that $\lambda_i < \lambda_j$ for $i > j$. Suppose that $\mathbf{L}(0) = \mathbf{O}$ and $\mathbf{w}_k^T(0)\mathbf{q}_k \neq 0$. A sufficient condition for KDR to hold for learning rule (8)+(12) is that $\psi_k(t) > y_k(t)z_k(t)$ at any time, and $\rho > 3/2$.*

*Proof.* The proof is carried out by induction, as in [5]. It is omitted from here because of space limitations. □

On the basis of Theorem 1, the choice of functions $\psi_k$'s may be illustrated. As exemplary discussion, we consider some cases for the class of functions of the form $\psi_k(t) = z_k(t)y_k(t) + \beta|y_k(t)|^\gamma$, where both $\gamma$ and $\beta$ are real positive constants, early mentioned in [8]. Namely, we consider the three cases $\gamma = 2$, $\gamma = 1$, $\beta = 0$.

- **Case I**: $\gamma = 2$, $\beta \neq 0$. Suppose the neuron $k^{\mathrm{th}}$ corresponds to a large eigenvalue $\lambda_k$ greater than 1. In this case, in the mean sense and after a suitable number of iterations, $y_k^2 > |y_k| > 0$, thus the fastest convergence speed is achieved for $\gamma = 2$; the corresponding algorithm is termed $y^2$–APEX. Note that $\psi_k = z_k y_k + \beta|y_k|^2$ closely recalls the term used in the original APEX algorithm.

- **Case II**: $\gamma = 1$, $\beta \neq 0$. Consider eigenvalues of $k^{\mathrm{th}}$ neuron be of medium size, i.e. $1 > \lambda_k > 0$. In this case, in the mean $|y_k| > y_k^2 > 0$, thus the choice $\gamma = 1$ allows for fast convergence. This case was first discussed and experimentally tested in [8], where the algorithm was termed $|y|$–APEX.

- **Case III**: $\beta = 0$. If the eigenvalue corresponding to the neuron is really small, i.e. $\lambda_k \approx 0$, there is no advantage in our model to choose $\psi_k$ different from $z_k y_k$ only, leading to the 0–APEX algorithm.

In [8] the computational complexity of the learning rules discussed in this paper has been estimated, as a function of network's size.

### 3.4. Possible generalizations

By properly adapting the employed optimization principle (i.e. by suitably selecting functions $C$ and $J$), it is possible to achieve two interesting generalizations. First, to make the algorithm able to perform PCA of complex-valued incoming data, as shown in [6]; second, to make it able to perform blind separation of circularly-distributed source signals; this involves both the extension to complex case and to non-quadratic (non-classical) optimization. Very preliminary results about this extension of old $\psi$–APEX have recently appeared in [7].

## 4. Experimental results

To show the behavior of the proposed class of APEX-like learning rules, several experiments have been performed. The aim of these experiments was to compare, in terms of performance on real-world signals, the proposed algorithms with the original APEX and the ones belonging to earlier $\psi$–APEX class. The signals considered consist of gray-scale images, at 8 bit per pixel. An $8 \times 8$ pixel window is slid without overlap from left to right and top to bottom of these images in order to extract the $p = 64$ inputs to the network. As a performance index, the Signal-to-Noise Ratio (SNR) between the reconstructed and the original image has been considered:

$$\mathrm{SNR} \stackrel{\mathrm{def}}{=} 10\log_{10}\frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c} I_{ij}^2}{\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}(I_{ij} - \bar{I}_{ij})^2} \ , \tag{13}$$

**Figure 2. The image 'child' used in the simulations.**

where $I_{ij}$ denotes the gray-value of $(i,j)^{\text{th}}$ pixel of the original $N_r \times N_c$ image whose principal components are to be extracted, and $\bar{I}_{ij}$ denotes the corresponding value of the reconstructed image. The step size $\eta$ has been set to $0.01$, and the magnifying factor $\rho$ has been set to $16$.

Using the $256 \times 256$ pixel image "child" (see Figure 2) a sequence of SNR values has been computed by sequentially extracting the first $m = 8$ principal components (PCs). The comparison among the APEX, with and without variable stepsize, to the improved $\psi$–APEX, shows that the latter ones exhibit better performances, as can be seen in the Figure 3, and that emphasizing the action of the inhibitory connections is a right way to improve the quality of principal component analysis. In particular, the experiments evidence how the improved $\psi$–APEX *do*

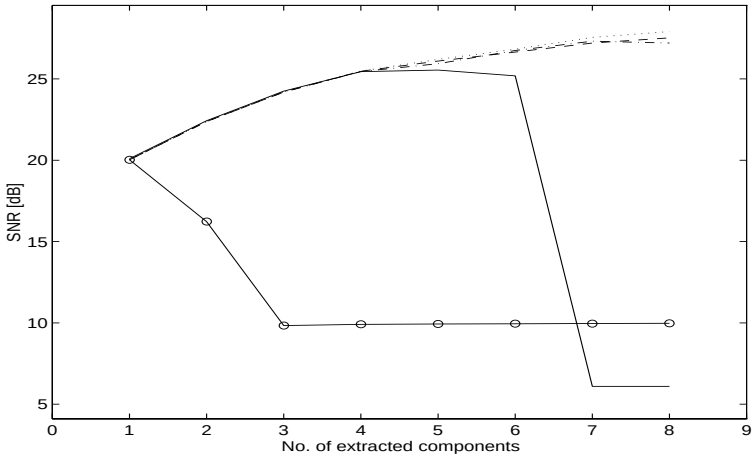


**Figure 3. Sequence of SNR values computed extracting sequentially 8 PCs from $8 \times 8$ image blocks (Solid-circle line: APEX; Solid line: variable stepsize APEX, Other lines: 0–APEX, $|y|$–APEX, $y^2$–APEX).**

*not need* a variable stepsize in order to achieve good results.

## 5. Conclusion

The aim of this paper was to present an improvement to $\psi$–APEX learnin rule class, relying on an optimization principle specialized for the laterally connected topology which emphasizes the importance of lateral inhibitory (decorrelating) connections. Through computer simulations performed on a digital image we showed that the proposed principal component algorithm exhibits better performances than original APEX one, without requiring a variable stepsize, thus keeping fairly limited the required computational efforts.

## References


[1] P.F. BALDI AND K. HORNIK, *Learning in neural networks: A survey*, IEEE Trans. on Neural Networks, Vol. 6, No. 4, pp. 837 – 858, July 1995

[2] S. BANNOUR AND M.R. AZIMI-SADJADI, *Principal Component extraction using recursive least squares learning*, IEEE Trans. on Neural Networks, Vol.6, No.2, March 1995

[3] A. CICHOCKI, W. KASPRZAK, AND W. SKARBEK, *Adaptive Learning Algorithm for Principal Component Analysis with Partial Data*, Proc. Cybernetics and Systems, Vol. 2, pp. 1014 – 1019, 1996

[4] K.I. DIAMANTARAS AND S.-Y. KUNG, *Cross-correlation neural network models*, IEEE Trans. on Signal Processing, Vol. 42, No. 11, Nov. 1994

[5] K.I. DIAMANTARAS AND S.-Y. KUNG, *Principal Component Neural Networks: Theory and Applications*, J. Wiley & Sons, 1996

[6] S. FIORI AND A. UNCINI, *A Unified Approach to Laterally-Connected Neural Nets*, Proc. of IX European Signal Processing Conference (EUSIPCO), Vol. I, pp. 379 – 382, 1998

[7] S. FIORI, A. UNCINI, AND F. PIAZZA, *Neural Blind Separation of Complex Sources by Extended APEX Algorithm (EAPEX)*, Proc. of International Symposium on Circuits and Systems, Vol. V, pp. 627 – 630, 1999

[8] S. FIORI *An Experimental Comparison of Three PCA Neural Networks*, Neural Processing Letters. Accepted for publication, expected to outcome on June, 2000

[9] S. HAYKIN, *Neural Networks*, Ed. MacMillan College Publishing Company, 1994

[10] J. KARHUNEN, *Optimization criteria and nonlinear PCA neural networks*, Proc. of International Joint Conference on Neural Networks (IEEE-IJCNN), pp. 1241 - 1246, 1994

[11] S.Y. KUNG, K.I. DIAMANTARAS AND J.S. TAUR, *Adaptive principal component extraction (APEX) and applications*, IEEE Trans. on Signal Processing, Vol.42, No.5, May 1994

[12] E. OJA, *Neural networks, principal components, and subspaces*, International Journal of Neural System, Vol. 1, pp. 61 – 68, 1989

[13] J. RUBNER AND P. TAVAN, *A self-organizing network for Principal-Component Analysis*, Europhysics Letters, Vol.10, No.7, pp. 693 – 698, 1989

[14] T.D. SANGER, *Optimal unsupervised learning in a single-layer linear feedforward neural network*, Neural Networks, Vol. 2, pp. 459 – 473, 1989

[15] A. WEINGESSEL AND K. HORNIK, *SVD algorithms: APEX-like versus Subspace Methods*, Neural Processing Letters, Vol. 5, pp. 177 – 184, 1997