CGHA for Principal Component Extraction in the Complex Domain

Yanwu Zhang, *Student Member, IEEE,* and Yuanliang Ma

***Abstract*—Principal component extraction is an efficient statistical tool which is applied to data compression, feature extraction, signal processing, etc. Representative algorithms in the literature can only handle real data. However, in many scenarios such as sensor array signal processing, complex data are encountered. In this paper, the complex domain generalized Hebbian algorithm (CGHA) is presented for complex principal component extraction. It extends the real domain generalized Hebbian algorithm (GHA) proposed by Sanger. Convergence of CGHA is analyzed. Like GHA, CGHA can be implemented by a single-layer linear neural network with simple computation. An example is given where CGHA is utilized in direction-of-arrival (DOA) estimation of multiple narrowband plane waves received by a sensor array.**

***Index Terms*— Complex domain, convergence, direction-of-arrival estimation, generalized Hebbian algorithm, neural network, principal component, single layer.**

I. INTRODUCTION

PRINCIPAL component extraction (or principal component analysis) [1]–[3] is a useful statistical tool for linearly reducing the dimensionality of a set of measurements while retaining as much information as possible [4]. This is accomplished by a linear mapping from the input space to a lower dimensional representation space [5].

Mathematically, principal component extraction carries out a linear transform from an N-dimensional zero-mean input vector space

$$X = [x_1 \quad x_2 \quad \cdots \quad x_N]^T \tag{1}$$

to an M-dimensional ($M < N$) output vector space

$$Y = [y_1 \; y_2 \; \cdots \; y_M]^T \tag{2}$$

and Y is related to X by

$$Y = W^H X \tag{3}$$

where W is an $N \times M$ matrix and its columns are the eigenvectors associated with the largest M eigenvalues of the input correlation matrix $R_{XX} = E[XX^H]$. T denotes transpose and H denotes conjugate transpose. The eigenvectors associated with the largest eigenvalues are called principal eigenvectors. Elements of vector Y are called principal components.

Manuscript received December 22, 1994; revised April 20, 1996 and April 2, 1997.

Y. Zhang is with the Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Y. Ma is with the College of Marine Engineering, Northwestern Polytechnic University, Xi'an 710072, China.

Publisher Item Identifier S 1045-9227(97)05248-X.

With $M < N$, the dimensionality of the input vector space is reduced. An N-dimensional "data space" is compressed into an M-dimensional "feature space." It can be proven [6] that principal component extraction is the optimal linear transform in the sense that it minimizes the least mean squared error when reconstructing X

$$X \approx WY. \tag{4}$$

Representative algorithms in the literature are for real data. In many scenarios however, input data are complex. Therefore it is necessary to extend the real domain algorithm to the complex domain.

In this paper, the complex domain generalized Hebbian algorithm (CGHA) is presented. It is an extension of the real domain principal component extraction algorithm, namely, generalized Hebbian algorithm (GHA) [1]. CGHA can be implemented with a single-layer linear neural network. Analysis of convergence of CGHA is given in the Appendix.

II. A BRIEF REVIEW OF GHA

The kernel of principal component extraction is to find principal eigenvectors. In 1989, Sanger presented the GHA [1]. With this algorithm, the principal eigenvectors of the input correlation matrix can be deduced iteratively with a single-layer linear neural network. Unlike with batch eigendecomposition, we need not compute the input correlation matrix in advance because the eigenvectors can be derived directly from the input data. Only local operations are called for and the neurons learn simultaneously. These features are attractive for parallel hardware realization. Successful applications in image coding and texture segmentation were carried out [1].

The mechanism of GHA can be summarized in the following.

The input column vector is

$$X = [x_1 \; x_2 \; \cdots x_N]^T. \tag{5}$$

In the decreasing order of eigenvalues, the M principal eigenvectors of the input correlation matrix $R_{XX} = E[XX^T]$ are expressed as the following column vectors:

$$W_1 = [w_{11} \; w_{12} \; \cdots \; w_{1N}]^T \tag{6}$$

$$W_2 = [w_{21} \; w_{22} \; \cdots \; w_{2N}]^T \tag{7}$$

$$\cdots$$

$$W_M = [w_{M1} \; w_{M2} \; \cdots \; w_{MN}]^T. \tag{8}$$

How to find $W_j (j = 1, 2, \cdots, M)$ is the crux of principal component extraction algorithms. In GHA, the initial value of

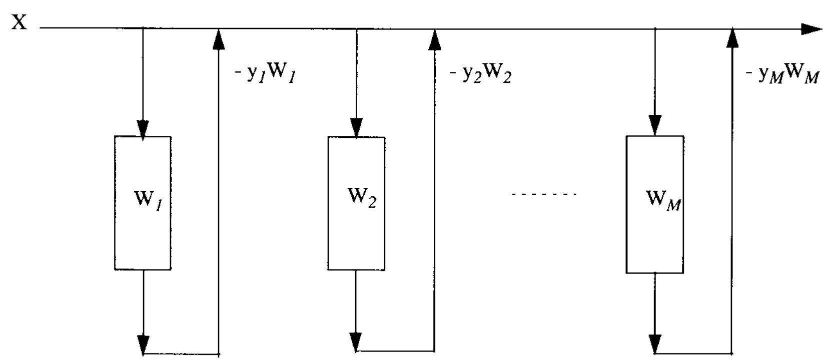


Fig. 1. Implementation network for GHA or CGHA.

$W_j (j = 1, 2, \cdots, M)$ can be randomly set [5]. The updating rule for W_j is

$$W_j(n+1) = W_j(n) + \mu(n) y_j(n) \cdot \left[X(n) - y_j(n) W_j(n) - \sum_{i<j} y_i(n) W_i(n) \right] \tag{9}$$

where n is the iteration index and

$$y_j(n) = W_j^T(n) X(n) \tag{10}$$

and $\mu(n)$ is the learning rate factor.

Sanger proved that W_j converges to the jth principal eigenvector of R_{XX}.

GHA can be implemented by a single-layer linear neural network, as shown in Fig. 1. Each block is a linear neuron.

Input vector $X = [x_1 \ x_2 \ \cdots \ x_N]^T$ is fed into M linear neurons, through an N-dimensional weight vector $W_j = [w_{j1} \ w_{j2} \ \cdots w_{jN}]^T$ to the jth neuron, $j = 1, 2, \cdots, M$. As the input vector flows through each neuron, $y_j W_j$ is subtracted from it successively as shown in (9). The output of the jth neuron is $y_j = W_j^T X$.

GHA is confined to the real domain. In many scenarios, we meet complex data. For example, in sensor array processing, input real data are usually transformed into complex data through quadrature sampling [7], [8], in order to utilize the narrowband phase-shift relationship between receptions of different sensors. Therefore, the complex version of GHA is of practical need.

III. Complex Domain Generalized Hebbian Algorithm

Now we present the CGHA. It is very similar to GHA except that complex notations are introduced. The updating rule for W_j is given by

$$W_j(n+1) = W_j(n) + \mu(n) \mathrm{conj}\,[y_j(n)] \cdot \left[X(n) - y_j(n) W_j(n) - \sum_{i<j} y_i(n) W_i(n) \right], \quad j = 1, 2, \cdots, M \tag{11}$$

where $\mathrm{conj}[y_j(n)]$ is the complex conjugate of

$$y_j(n) = W_j^H(n) X(n) \tag{12}$$

where H denotes Hermitian transpose and $\mu(n)$ is the learning rate factor.

In the Appendix, we show that with any initial W_j, it converges to the jth normalized eigenvector of $R_{XX} = E[XX^H]$.

Comparing (11) and (12) of CGHA with (9) and (10) of GHA, we find that GHA is a simplified version of CGHA.

The implementation network for CGHA is exactly the same as that for GHA: a single-layer linear neural network as shown in Fig. 1.

Like GHA, CGHA possesses the following features.

1) No need to compute the correlation matrix R_{XX} in advance. The eigenvectors are derived (learned) directly from the input vector. In sensor array processing, the input vector is one "snapshot" of all sensor receptions at one temporal sampling. When the number of sensors is large, this advantage becomes significant because computation of R_{XX} is time consuming.
2) Implementation with local operation. This feature is favorable for parallel hardware. Equation (11) can be rewritten as

$$\Delta W_j = \mu \,\mathrm{conj}\,(y_j)[X_j - y_j W_j] \tag{13}$$

where X_j means the "net" input to the jth neuron: at the nth iteration, the net input to no. 1 neuron is $X(n)$; that to no. 2 neuron is $X(n) - y_1(n)W_1(n)$; $\cdots$ that to no. M neuron is $X(n) - y_1(n)W_1(n) - y_2(n)W_2(n) - \cdots - y_{M-1}(n)W_{M-1}(n)$, i.e., subtracting $y_k(n)W_k(n)$ from $X(n)$ progressively as it goes from no. k neuron to No. $(k+1)$ neuron. Considering "net" input to each neuron, weight updating is local.
3) Good expandability. Updating of the jth neuron is affected only by those neurons with number less than j. Hence, if the first M neurons have already converged, i.e., the first M principal eigenvectors have been obtained, then the learning of the $(M+1)$th neuron will leave intact the preceding M neuron weight vectors.

IV. Application of CGHA to DOA Estimation

Assume a uniform linear array composed of N sensors with identical directivity. D narrowband signals impinge on the array as plane waves from directions $\theta_1, \theta_2, \cdots, \theta_D$. Suppose the received noise is spatially white with zero mean and variance σ^2. The received complex data vector can be expressed as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{bmatrix} = [a(\theta_1)\ a(\theta_2)\ \cdots\ a(\theta_D)] \begin{bmatrix} F_1 \\ F_2 \\ \cdots \\ F_D \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_N \end{bmatrix} \quad (14)$$

where

$$a(\theta_i) = \begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_i/\lambda_i)} \\ \cdots \\ e^{j2\pi(N-1)(d \sin \theta_i/\lambda_i)} \end{bmatrix} \qquad i = 1, 2, \cdots, D \quad (15)$$

is the steering vector of the ith signal source with incident angle θ_i.

F_i is the ith signal, λ_i is the wavelength of the ith signal, and n_j is the noise received by the jth sensor.

As long as $a[\theta_i]\,(i = 1, 2, \cdots, D)$ are obtained, the directions of impinging signals are found.

It can be proven [9] that for the input correlation matrix $R_{XX} = E[XX^H]$, those eigenvectors associated with eigenvalues greater than σ^2 are linear combinations of $a(\theta_i)$, i.e.,

$$W_k = \sum_{i=1}^{D} \alpha_{ki} a(\theta_i) \qquad k = 1, 2, \cdots, M \text{ where } M \le D. \quad (16)$$

Therefore, the principal eigenvectors contain information of source directions. With (15) plugged in (16), we have a clearer observation of W_k

$$W_k = \alpha_{k1} \begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_1/\lambda_1)} \\ \cdots \\ e^{j2\pi(N-1)(d \sin \theta_1/\lambda_1)} \end{bmatrix} + \alpha_{k2} \begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_2/\lambda_2)} \\ \cdots \\ e^{j2\pi(N-1)(d \sin \theta_2/\lambda_2)} \end{bmatrix} + \cdots + \alpha_{kD} \begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_D/\lambda_2)} \\ \cdots \\ e^{j2\pi(N-1)(d \sin \theta_D/\lambda_2)} \end{bmatrix}. \quad (17)$$

If we deem column vector

$$\begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_i/\lambda_i)} \\ \cdots \\ e^{j2\pi(N-1)(d \sin \theta_i/\lambda_i)} \end{bmatrix}$$

as a sinusoid of frequency $2\pi(d \sin \theta_i/\lambda_i)$, then W_k can be deemed as a one-dimensional sequence composed of multiple

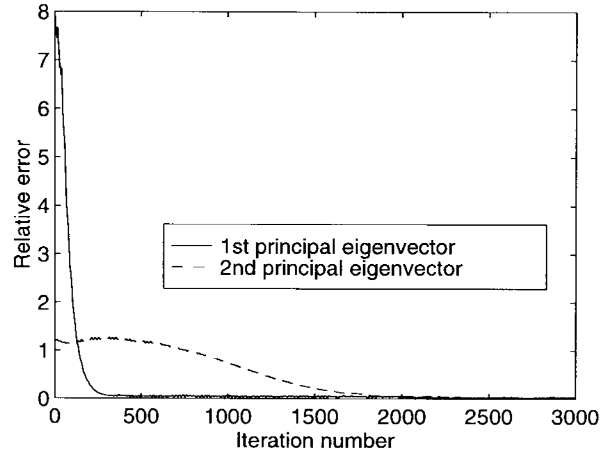


Fig. 2. Learning curves for the first and the second principal eigenvectors.

sinusoids. By finding the frequency components, $\theta_i\,(i = 1, 2, \cdots, D)$ are obtained. With this approach [10], the original problem with spatio-temporal coupling is reduced to an easier problem of one-dimensional frequency analysis. Moreover, this direction estimation method works well whether or not the signal sources are correlated.

The key step is to derive principal eigenvectors of the input correlation matrix. The data are complex, so CGHA can be employed.

The following is a simulation of DOA estimation using CGHA. Two of the three signals are coherent since they are of the same frequency. For some popular high-resolution DOA estimators such as MUSIC, the two coherent signals cannot be resolved unless spatial smoothing is conducted at the cost of effective array aperture [11]. With the method described in this section, however, all the three signals are resolved without resorting to spatial smoothing.

Consider a 15-sensor uniform linear array. Two coherent signals and one incoherent signal are received. Their parameters are as follows.

Normalized frequencies (relative to sampling frequency) $f_1 = f_2 = 0.2$, $f_3 = 0.15$. Incident angles $\theta_1 = 10°$, $\theta_2 = 40°$, $\theta_3 = 35°$. The sensor spacing is $d = \frac{1}{2}\lambda_1 = \frac{1}{2}\lambda_2 = \frac{3}{8}\lambda_3$ where λ_i is the wavelength. SNR of signals are 20 dB for the first and 14 dB for the second and third.

The simultaneous learning curves for the first and the second eigenvectors are shown in Fig. 2. The relative error is defined as $\|W_{k,precise} - W_{k,\text{CGHA}}\|_2/\|W_{k,precise}\|_2$ for the kth principal eigenvector, where $W_{k,precise}$ is the precise eigenvector. $W_{k,\text{CGHA}}$ is the eigenvector learned by CGHA. $\|\cdot\|_2$ denotes Euclidean norm. After 3000 iterations, the relative errors are 4.6 and 2.1% for the first and the second principal eigenvectors, respectively.

Using AR modeling to analyze the principal eigenvectors obtained with CGHA, we can get high-resolution spectrum showing source directions. The spectrum of the second principal eigenvector is shown in Fig. 3. The three peaks lie at $\phi_1 = 2\pi \cdot 0.085$, $\phi_2 = 2\pi \cdot 0.329$, $\phi_2 = 2\pi \cdot 0.215$. With formula $\phi_i = 2\pi(d \sin \theta_i/\lambda_i)$ we get the estimations of DOA: $\theta_1 = 9.8°$, $\theta_2 = 41.1°$, $\theta_3 = 35.0°$ which are very close to true values.

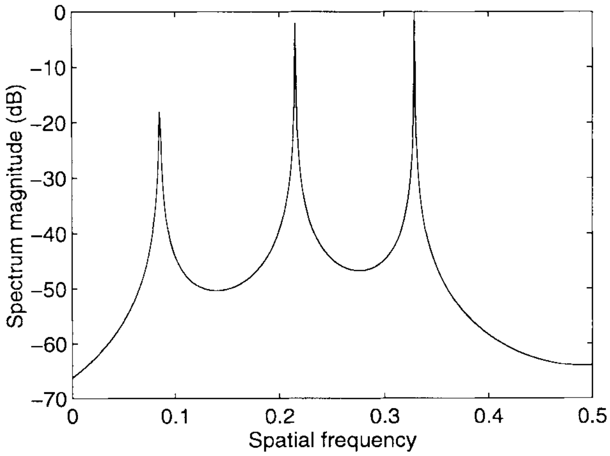


Fig. 3. AR spectrum of the second principal eigenvector.

V. Conclusion

CGHA is presented in this paper and its convergence is analyzed. This complex domain algorithm can be realized by a single-layer linear neural network. It possesses features attractive for practical implementation: no need to compute the input correlation matrix, local operation, good expandability, etc. When eigendecomposition, data compression, or feature extraction for complex data is needed, CGHA can play an efficient role.

An application of CGHA to sensor array signal processing is demonstrated. Converged principal eigenvectors are obtained and directions of signal sources are well estimated.

Appendix
Convergence Analysis of CGHA

The convergence analysis of CGHA extends Sanger's analysis on GHA [1] to the complex domain. We rewrite CGHA algorithm in matrix form to include all M principal eigenvectors

$$\begin{aligned} W(n+1) =& W(n) \\ &+ \mu(n)\{X(n)X^H(n)W(n) \\ &- W(n)UT[Y(n)(Y^H\{n\})]\} \end{aligned} \tag{A.1}$$

where $W = [W_1 \;\; W_2 \;\; \cdots \;\; W_M]$ is an $N \times M$ matrix composed of column vectors W_j, $j = 1, 2, \cdots, M$. $Y(n) = W^H(n)X(n)$.UT$[\cdot]$ sets all elements below the diagonal of the square matrix to zero, thereby producing an upper triangular (UT) matrix.

Taking expectation on both sides of (A.1) and noticing that $R_{XX}(n) = E[X(n)X^H(n)]$, we have

$$\begin{aligned} W(n+1) =& W(n) \\ &+ \mu(n)\{R_{XX}(n)W(n) \\ &- W(n)\mathrm{UT}[W^H(n)R_{XX}(n)W(n)]\}. \end{aligned} \tag{A.2}$$

The convergence property for the above difference equation is the same as that for the following differential equation:

$$\begin{aligned} \frac{d}{dt}W(t) =& R_{XX}(t)W(t) \\ &- W(t)\mathrm{UT}[W^H(t)R_{XX}(t)W(t)]. \end{aligned} \tag{A.3}$$

In the following, we analyze the convergence in two steps.

1) W_1 converges to the eigenvector associated with the largest eigenvalue.

W_1 is the first column of matrix W. According to (A.3), its evolution is governed by

$$\begin{aligned} \frac{d}{dt}W_1(t) =& R_{XX}(t)W_1(t) \\ &- W_1(t)[W_1^H(t)R_{XX}(t)W_1(t)]. \end{aligned} \tag{A.4}$$

Assume R_{XX} is positive definite with N distinct eigenvalues $\lambda_1 > \lambda_2 > \cdots \lambda_N$ which are associated with corresponding orthonormalized eigenvectors $e_1, e_2, \cdots, e_N$. (Cases of repeated or zero eigenvalues are straightforward generalizations.) Note that since R_{XX} is Hermitian, all of its eigenvalues are real.

Expand W_1 in terms of the entire orthonormal set of eigenvectors as

$$W_1 = \sum_{k=1}^{N} c_k e_k \tag{A.5}$$

where $c_k = e_k^H W_1$. Plugging (A.5) together with $R_{XX}e_k = \lambda_k e_k$ into (A.4) gives

$$\begin{aligned} \sum_{k=1}^{N} \frac{dc_k}{dt} e_k =& \sum_{k=1}^{N} c_k \lambda_k e_k \\ &- \left(\sum_{l=1}^{N} |c_l|^2 \lambda_l\right) \sum_{k=1}^{N} c_k e_k \end{aligned} \tag{A.6}$$

where $|\cdot|$ denotes norm of a complex variable.

Premultiply e_k^H to both sides of (A.6), and the orthonormality of $\{e_k\}$ leads to

$$\frac{dc_k}{dt} = c_k\left(\lambda_k - \sum_{l=1}^{N} |c_l|^2 \lambda_l\right) \tag{A.7}$$

1.1) For $k > 1$.

Define $r_k = c_k/c_1$ (assume $c_1 \neq 0$), and then we have

$$\frac{dr_k}{dt} = \left(\frac{1}{c_1}\right)\left(\frac{dc_k}{dt} - r_k\frac{dc_1}{dt}\right) \tag{A.8}$$

Using (A.7), we have

$$\begin{aligned} \frac{dr_k}{dt} =& \left(\frac{1}{c_1}\right)\Bigg[c_k\left(\lambda_k - \sum_{l=1}^{N} |c_l|^2 \lambda_l\right) \\ &- r_k c_1\left(\lambda_1 - \sum_{l=1}^{N} |c_l|^2 \lambda_l\right)\Bigg] \end{aligned} \tag{A.9}$$

which is simplified to

$$\frac{dr_k}{dt} = r_k(\lambda_k - \lambda_1). \tag{A.10}$$

The solution to the above differential equation is

$$r_k(t) = r_k(0)\exp[(\lambda_k - \lambda_1)t]. \tag{A.11}$$

However, $\lambda_k < \lambda_1$ for any $k > 1$. Therefore, $r_k(t)$ exponentially decays to zero with any $r_k(0)$, i.e., $r_k \to 0$ for $k > 1$.

1.2) For $k = 1$

$$\frac{dc_1}{dt} = c_1\left(\lambda_1 - |c_1|^2\lambda_1 - |c_1|^2\sum_{l=2}^{N}|r_l|^2\lambda_l\right). \quad \text{(A.12)}$$

Assume t is large, so r_l for $l > 1$ is negligible. Hence we drop the last term, and (A.12) becomes

$$\frac{dc_1}{dt} = c_1(\lambda_1 - |c_1|^2\lambda_1). \quad \text{(A.13)}$$

To show that $c_1(t)$ converges, we define another function

$$V = (|c_1|^2 - 1)^2. \quad \text{(A.14)}$$

Utilizing (A.13), we have

$$\frac{dV}{dt} = -4\lambda_1|c_1|^2(|c_1|^2 - 1)^2. \quad \text{(A.15)}$$

So we see that $dV/dt \le 0$. Thus V is a Lyapunov function and takes its minimum at $|c_1| = 1$. Therefore, $|c_1(t)| \to 1$ with any $c_1(0)$.

In 1.1) it is shown that $r_k \to 0$ for $k > 1$. In 1.2) it is shown that $|c_1| \to 1$. We know that $W_1 = c_1e_1 + c_1\sum_{k=2}^{N} r_ke_k$. Therefore the last term decays to zero. For any initial value $W_1(0)$, $W_1(t) \to e_1$ with a complex factor of norm one.

2) For $j > 1$, W_j converges to the eigenvector associated with the jth largest eigenvalue.

We use induction to show that if the first j-1 columns of matrix W converge to the first j-1 principal eigenvectors, then the jth column W_j will converge to the jth principal eigenvector.

The evolution of W_j is governed by

$$\frac{d}{dt}W_j(t) = R_{XX}(t)W_j(t) - \sum_{k\le j} W_k(t)[W_k^H(t)R_{XX}(t)W_j(t)]. \quad \text{(A.16)}$$

At time t, we can express W_k as

$$W_k(t) = e_k + \varepsilon_k(t)f_k(t) \quad \text{(A.17)}$$

where e_k is the kth normalized eigenvector of R_{XX}; f_k is a time-varying unit-length vector; ε_k is a scalar.

Based on the premise of the induction, we know that for $k < j$, $\varepsilon_k(t) \to 0$.

Combining (A.16) and (A.17) gives

$$\begin{aligned}\frac{d}{dt}W_j(t) = {} & R_{XX}(t)W_j(t) \\ & - W_j(t)[W_j^H(t)R_{XX}(t)W_j(t)] \\ & - \sum_{k<j} e_k[e_k^H R_{XX}(t)W_j(t)] \\ & + O(\varepsilon) + O(|\varepsilon|^2)\end{aligned} \quad \text{(A.18)}$$

where ε indicates a term converging to zero at least as fast as the slowest decaying ε_k for $k < j$.

Assuming time is large, we neglect term $O(\varepsilon)$ and $O(|\varepsilon|^2)$.

Expand W_j in terms of the entire orthonormal set of eigenvectors as $W_j = \sum_{k=1}^{N} c_ke_k$, where $c_k = e_k^HW_j$. Plugging this expansion together with $R_{XXe_k} = \lambda_ke_k$ into (A.18) gives

$$\begin{aligned}\sum_{k=1}^{N}\frac{dc_k}{dt}e_k = {} & -\sum_{k<j}\left(\sum_{l=1}^{N}|c_l|^2\lambda_l\right)c_ke_k \\ & + \sum_{k=j}^{N}\left(\lambda_k - \sum_{l=1}^{N}|c_l|^2\lambda_l\right)c_ke_k.\end{aligned} \quad \text{(A.19)}$$

Premultiplying e_k^H to both sides of (A.19), and utilizing the orthonormality of $\{e_k\}$, we have

$$\frac{dc_k}{dt} = -c_k\sum_{l=1}^{N}|c_l|^2\lambda_l \qquad \text{for } k < j \quad \text{(A.20)}$$

$$\frac{dc_k}{dt} = c_k\left(\lambda_k - \sum_{l=1}^{N}|c_l|^2\lambda_l\right) \qquad \text{for } k \ge j. \quad \text{(A.21)}$$

2.1) For $k < j$

The solution to the differential equation is

$$c_k(t) = c_k(0)\exp\left[-\left(\sum_{l=1}^{N}|c_l|^2\lambda_l\right)t\right]. \quad \text{(A.22)}$$

R_{XX} is positive definite, so $\lambda_l > 0$. Thus $-(\sum_{l=1}^{N}|c_l|^2\lambda_l) < 0$. $c_k(t)$ exponentially decays to zero with any $c_k(0)$, i.e., $c_k(t) \to 0$ for $k < j$.

2.2) For $k > j$

Define $r_k = c_k/c_j$ (assume $c_j \ne 0$), and then we have

$$\frac{dr_k}{dt} = \left(\frac{1}{c_j}\right)\left(\frac{dc_k}{dt} - r_k\frac{dc_j}{dt}\right). \quad \text{(A.23)}$$

Using (A.21), we have

$$\begin{aligned}\frac{dr_k}{dt} = \left(\frac{1}{c_j}\right)\Bigg[& c_k\left(\lambda_k - \sum_{l=1}^{N}|c_l|^2\lambda_l\right) \\ & - r_kc_j\left(\lambda_j - \sum_{l=1}^{N}|c_l|^2\lambda_l\right)\Bigg]\end{aligned} \quad \text{(A.24)}$$

which is simplified to

$$\frac{dr_k}{dt} = r_k(\lambda_k - \lambda_j). \quad \text{(A.25)}$$

The solution to the above differential equation is

$$r_k(t) = r_k(0)\exp[(\lambda_k - \lambda_j)t]. \quad \text{(A.26)}$$

However, $\lambda_k < \lambda_j$ for any $k > j$. Therefore, $r_k(t)$ exponentially decays to zero with any $r_k(0)$, i.e., $r_k(t) \to 0$ for $k > j$.

2.3) For $k = j$

$$\frac{dc_j}{dt} = c_j\left(\lambda_j - |c_j|^2\lambda_j - |c_j|^2\sum_{l>j}|r_l|^2\lambda_l - \sum_{l<j}|c_l|^2\lambda_l\right). \quad \text{(A.27)}$$

Assume t is large. It has been shown in 2.1) that $c_l \to 0$ for $l < j$ and $r_l \to 0$ for $l > j$. Hence we drop the last two terms, and the equation becomes

$$\frac{dc_j}{dt} = c_j(\lambda_j - |c_j|^2 \lambda_j). \tag{A.28}$$

To show that $c_j(t)$ converges, we define another function

$$P = (|c_j|^2 - 1)^2. \tag{A.29}$$

Utilizing (A.28), we have

$$\frac{dP}{dt} = -4\lambda_j |c_j|^2 (|c_j|^2 - 1)^2. \tag{A.30}$$

So we see that $dP/dt \le 0$. Thus P is a Lyapunov function and takes its minimum at $|c_j| = 1$. Therefore, $|c_j(t)| \to 1$ with any $c_j(0)$.

In 2.1) it is shown that $c_k \to 0$ for $k < j$. In 2.2) it is shown that $r_k \to 0$ for $k > j$. in 2.3) it is shown that $|c_j(t)| \to 1$. We know that $W_j = c_j e_j + \sum_{k<j} c_k e_k + c_j \sum_{k>j} r_k e_k$. Therefore the last two terms decay to zero. For any initial value $W_j(0)$, $W_j(t) \to e_j$ with a complex factor of norm one.

With the above analyzes of 1) and 2), we arrive at the conclusion that columns of matrix W converge to corresponding eigenvectors of R_{XX}. In other words, CGHA converges.

ACKNOWLEDGMENT

The authors are grateful to the editors and reviewers for their helpful comments and suggestions.

REFERENCES

[1] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks,* vol. 2, pp. 459–473, 1992.

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

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation.* New York: Macmillan, 1994.

[4] S. Bannour and M. Azimi-sadjadi, "Principal component extraction using recursive least squares learning," *IEEE Trans. Neural Networks,* vol. 6, pp. 457–469, 1995.

[5] M. Plumbley, "Lyapunov functions for convergence of principal component algorithms," *Neural Networks,* vol. 8, no. 1, pp. 11–23, 1995.

[6] K. Hornik and C. Kuan, "Convergence analysis of local feature extraction algorithms," *Neural Networks,* vol. 5, pp. 229–240, 1992.

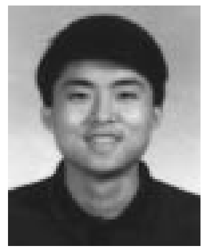
[7] R. Nielsen, *Sonar Signal Processing.* Norwood, MA: Artech House, 1991.

[8] Y. Zhang and Y. Ma, "An efficient architecture for real-time narrowband beamforming," *IEEE J. Oceanic Eng.,* vol. 19, pp. 635–638, 1994.

[9] J. A. Cadzow, "A high resolution direction-of-arrival algorithm for narrow-band coherent and incoherent sources," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-36, pp. 965–979, July 1988.

[10] Y. Zhang and Y. Ma, "Estimating direction of arrival with one-dimensional spectral analysis," *Proc. MTS/IEEE Oceans' 95,* San Diego, CA, Oct. 1995, pp. 845–848.

[11] T. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, pp. 806–811, 1985.



Yanwu Zhang (S'95) was born in June 1969 in Shaanxi Province, China. He received the B.S. degree in electrical engineering and M.S. degree in acoustic signal processing from Northwestern Polytechnic University, Xi'an, China, in 1989 and 1991, respectively.

Before 1994, his research was focused on sensor array beamforming and bearing estimation algorithms, and their digital signal processor (DSP) implementation. From 1994 to 1995, he was a Research Assistant at the School of Oceanography of University of Washington, Seattle, WA, studying oceanography and seismology. From September to December 1995, he was a Teaching Assistant at the Department of Ocean Engineering of Massachusetts Institute of Technology, Cambridge. Since the beginning of 1996, he has been a Research Assistant at the Autonomous Underwater Vehicle (AUV) Laboratory of MIT Sea Grant. His current research interests are AUV survey design and implementation, oceanographic instrumentation and modeling, adaptive signal processing, and neural networks.

Mr. Zhang is a Student Member of SNAME (Society of Naval Architects and Marine Engineers).

Yuanliang Ma was born in 1938 in Sichuan Province, China. He received B.Sc. degree in underwater acoustics from Northwestern Polytechnical University (NPU), Xi'an, China, in 1961.

Since then he has been working on underwater acoustics and signal processing, mainly at NPU. From 1981 to 1983, he was a Visiting Scholar with the Department of Electrical and Electronic Engineering, Loughborough University of Technology, England. He then became an Associate Professor of NPU in 1980 and Full Professor in 1985. He has published books on underwater acoustic transducers, and on adaptive active noise control in addition to about 150 journal and conference papers. His current research interests are in sensor array processing, matched field processing and ocean acoustic tomography, neural network for signal processing, and ocean sediment classification.

Mr. Ma is now a standing Councilor of the Acoustical Society of P. R. China, as well as the Chairman of the Youth Chapter of that society. He has also received a dozen awards for achievements in scientific and technological progress from the state and ministries.