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

\[ X = [x_1 \; x_2 \; \cdots \; x_N]^T \]  

(1)

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

\[ Y = [y_1 \; y_2 \; \cdots \; y_M]^T \]  

(2)

and \( Y \) is related to \( X \) by

\[ Y = W^H X \]  

(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.

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 \( X \)

\[ X \approx WY. \]  

(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

\[ X = [x_1 \; x_2 \; \cdots \; x_N]^T. \]  

(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:

\[ W_1 = [w_{11} \; w_{12} \; \cdots \; w_{1M}]^T \]  

(6)

\[ W_2 = [w_{21} \; w_{22} \; \cdots \; w_{2M}]^T \]  

(7)

\[ \cdots \]  

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

(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

$$W_j(n + 1) = W_j(n) + \mu(n)y_j(n)$$

$$\cdot \left[ X(n) - y_j(n)W_j(n) - \sum_{i < j} y_i(n)W_i(n) \right]$$

where $n$ is the iteration index and

$$y_j(n) = W_j^H(n)X(n)$$

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

Sanger proved that $W_j$ converges to the $j$th 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 $j$th neuron, $j = 1, 2, \cdots, M$. As the input vector flows through each neuron, $y_jW_j$ is subtracted from it successively as shown in (9). The output of the $j$th neuron is $y_j = W_j^T(n)X(n)$.

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

$$W_j(n + 1) = W_j(n) + \mu(n)\text{co}n_j[y_j(n)]$$

$$\cdot \left[ X(n) - y_j(n)W_j(n) - \sum_{i < j} y_i(n)W_i(n) \right]$$

where $\text{co}n_j[y_j(n)]$ is the complex conjugate of $y_j(n) = W_j^H(n)X(n)$ 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 $j$th 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

$$\Delta W_j = \mu(n)\overline{y_j}X_j - y_jW_j$$

where $X_j$ means the “net” input to the $j$th neuron: at the $r$th 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 $j$th 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 $\sigma^2$. The received complex data vector can be expressed as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = [\alpha(\theta_1) \ a(\theta_2) \ \cdots \ a(\theta_D)] \begin{bmatrix} F_1 \\ F_2 \\ \cdots \\ F_D \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_N \end{bmatrix}$$  \hspace{1cm} (14)

where

$$\alpha(\theta_i) = \begin{bmatrix} e^{j2\pi(d \sin \theta_i/\lambda_1)} \\ \vdots \\ e^{j2\pi(N-1)(d \sin \theta_i/\lambda_1)} \end{bmatrix} \quad i = 1, 2, \cdots, D$$  \hspace{1cm} (15)

is the steering vector of the $i$th signal source with incident angle $\theta_i$. $F_i$ is the $i$th signal, $\lambda_i$ is the wavelength of the $i$th signal, and $n_j$ is the noise received by the $j$th sensor.

As long as $\alpha(\theta_i)$ ($i = 1, 2, \cdots, D$) 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 $\sigma^2$ are linear combinations of $\alpha(\theta_i)$, i.e.,

$$W_k = \sum_{i=1}^{D} \alpha_k \alpha(\theta_i) \quad k = 1, 2, \cdots, M \text{ where } M \leq D.$$  \hspace{1cm} (16)

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

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

If we deem column vector

$$\begin{bmatrix} 1 \\ e^{j2\pi(d \sin \theta_1/\lambda_1)} \\ \vdots \\ e^{j2\pi(N-1)(d \sin \theta_1/\lambda_1)} \end{bmatrix}$$

as a sinusoid of frequency $2\pi(d \sin \theta_1/\lambda_1)$, then $W_k$ can be deemed as a one-dimensional sequence composed of multiple 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.

Normalised frequencies (relative to sampling frequency) $f_1 = f_2 = 0.2, f_3 = 0.15$. Incident angles $\theta_1 = 10^\circ, \theta_2 = 40^\circ, \theta_3 = 35^\circ$. The sensor spacing is $d = \frac{1}{2} \lambda_3 = \frac{1}{2} \lambda_2 = \frac{3}{8} \lambda_3$ where $\lambda_3$ 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 $||W_{k,\text{precise}} - W_{k,\text{CGHA}}||2/||W_{k,\text{precise}}||2$ for the $k$th principal eigenvector, where $W_{k,\text{precise}}$ 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: $\theta_1 = 9.8^\circ, \theta_2 = 41.1^\circ, \theta_3 = 35.0^\circ$ which are very close to true values.
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

\[
W(n+1) = W(n) + \mu(n)\{X(n)X^H(n)W(n) - \mu(n)UTY(n)Y^H\{n\}\}, \quad (A.1)
\]

where \( W = [W_1 \ W_2 \ \cdots \ W_M] \) is an \( N \times M \) matrix composed of column vectors \( W_j, \ j = 1, 2, \cdots, M \). \( Y(n) = W^H(n)X(n)UT[n] \) 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

\[
W(n+1) = W(n) + \mu(n)\{R_{XX}(n)W(n) - W(n)UT[W^H(n)R_{XX}(n)W(n)]\}, \quad (A.2)
\]

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

\[
\frac{d}{dt} W(t) = R_{XX}(t)W(t) - W(t)UT[W^H(t)R_{XX}(t)W(t)], \quad (A.3)
\]

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

\[
\frac{d}{dt} W_1(t) = R_{XX}(t)W_1(t) - W_1(t)[W_1^H(t)R_{XX}(t)W_1(t)]. \quad (A.4)
\]

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

\[
W_1 = \sum_{k=1}^{N} c_k e_k \quad (A.5)
\]

where \( c_k = e_k^HW_1 \). Plugging (A.5) together with \( R_{XX}c_k = \lambda_k c_k \) into (A.4) gives

\[
\sum_{k=1}^{N} \frac{dc_k}{dt} c_k = \sum_{k=1}^{N} c_k \lambda_k c_k - \left( \sum_{k=1}^{N} |c_k|^2 \lambda_k \right) \sum_{k=1}^{N} c_k e_k \quad (A.6)
\]

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

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

1.1) For \( k > 1 \).

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

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

Using (A.7), we have

\[
\frac{dr_k}{dt} = \left( \frac{1}{c_1} \right) \left( c_k \left( \lambda_k - \sum_{l=1}^{N} |c_l|^2 \lambda_l \right) - r_k c_1 \left( \lambda_1 - \sum_{l=1}^{N} |c_l|^2 \lambda_l \right) \right) \quad (A.9)
\]

which is simplified to

\[
\frac{dr_k}{dt} = r_k (\lambda_k - \lambda_1). \quad (A.10)
\]

The solution to the above differential equation is

\[
r_k(t) = r_k(0) \exp[(\lambda_k - \lambda_1)t]. \quad (A.11)
\]

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 $k = 1$

\[
\frac{d c_1}{dt} = c_1 \left( \lambda_1 - \left| c_1 \right|^2 \lambda_1 - \sum_{l=2}^{N} \left| \eta_l \right|^2 \lambda_l \right). \tag{A.12}
\]

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

\[
\frac{d c_1}{dt} = c_1 (\lambda_1 - \left| c_1 \right|^2 \lambda_1). \tag{A.13}
\]

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

\[
V = (\left| c_1 \right|^2 - 1)^2. \tag{A.14}
\]

Utilizing (A.13), we have

\[
\frac{dV}{dt} = -4\lambda_1 \left| c_1 \right|^2 (\left| c_1 \right|^2 - 1)^2. \tag{A.15}
\]

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

In 1.1) it is shown that $\tau_k \to 0$ for $k > 1$. In 1.2) it is shown that $\left| c_1 \right| \to 1$. We know that $W_1 = C_1 = c_1 + \sum_{l=2}^{N} \tau_l c_k$. Therefore the last term decays to zero. For any initial value $W_1(0)$, $W_1(t) \to c_1$ with a complex factor of norm one.

2) For $j > 1$. $W_j$ converges to the eigenvector associated with the $j$th 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 $j$th column $W_j$ will converge to the $j$th principal eigenvector.

The evolution of $W_j$ is governed by

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

At time $t$, we can express $W_k$ as

\[
W_k(t) = c_k + \varepsilon_k(t)f_k(t) \tag{A.17}
\]

where $c_k$ is the $k$th normalized eigenvector of $R_{XX}$; $f_k$ is a time-varying unit-length vector; $\varepsilon_k$ 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

\[
\frac{d}{dt} W_j(t) = R_{XX}(t)W_j(t)
- W_j(t)W_j^H(t)R_{XX}(t)W_j(t)
- \sum_{k < j} c_k f_k^H(t)R_{XX}(t)W_j(t)
+ O(\varepsilon) + O(|\varepsilon|^2) \tag{A.18}
\]

where $\varepsilon$ indicates a term converging to zero at least as fast as the slowest decaying $\varepsilon_k$ 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 $W_j = \sum_{k=1}^{N} c_k c_k$, where $c_k = f_k^H W_j$.

Plugging this expansion together with $R_{XX} c_k = \lambda_k c_k$ into (A.18) gives

\[
\sum_{k=1}^{N} \frac{dc_k}{dt} c_k = -\sum_{k < j} \left( \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \right) c_k c_k
+ \sum_{k = j}^{N} \left( \lambda_k - \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \right) c_k c_k. \tag{A.19}
\]

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

\[
\frac{dc_k}{dt} = -c_k \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \quad \text{for } k < j \tag{A.20}
\]

\[
\frac{dc_k}{dt} = c_k \left( \lambda_k - \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \right) \quad \text{for } k \geq j. \tag{A.21}
\]

2.1) For $k < j$

The solution to the differential equation is

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

$R_{XX}$ is positive definite, so $\lambda_k > 0$. Thus $-(\sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l) < 0$. $c_k(t)$ 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 $\tau_k = c_k / c_j$ (assume $c_j \neq 0$), and then we have

\[
\frac{d\tau_k}{dt} = \left( \frac{1}{c_j} \right) \left( \frac{dc_k}{dt} - \tau_k \frac{dc_j}{dt} \right). \tag{A.23}
\]

Using (A.21), we have

\[
\frac{d\tau_k}{dt} = \left( \frac{1}{c_j} \right) \left[ c_k \left( \lambda_k - \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \right)
- \tau_k c_j \left( \lambda_j - \sum_{l=1}^{N} \left| q_l \right|^2 \lambda_l \right) \right] \tag{A.24}
\]

which is simplified to

\[
\frac{d\tau_k}{dt} = \tau_k (\lambda_k - \lambda_j). \tag{A.25}
\]

The solution to the above differential equation is

\[
\tau_k(t) = \tau_k(0) \exp \left[ (\lambda_k - \lambda_j) t \right]. \tag{A.26}
\]

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

2.3) For $k = j$

\[
\frac{dc_j}{dt} = c_j \left( \lambda_j - \left| c_j \right|^2 \lambda_j - \sum_{l > j} \left| q_l \right|^2 \lambda_l - \sum_{k < j} \left| q_k \right|^2 \lambda_k \right). \tag{A.27}
\]
Assume $\tau$ is large. It has been shown in 2.1) that $q_l \to 0$ for $l < j$ and $r_k \to 0$ for $l > j$. Hence we drop the last two terms, and the equation becomes

$$\frac{dc_j}{dt} = c_j(\lambda_j - |c_j|^2 \lambda_j).$$  \hspace{1cm} (A.28)

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

$$P = (|c_j|^2 - 1)^2.$$

(A.29)

Utilizing (A.28), we have

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

(A.30)

So we see that $dP/dt \leq 0$. 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 $W_j = c_j c_j^T + \sum_{k < j} c_k c_k^T + c_j \sum_{k > j} r_k c_k^T$. Therefore the last two terms decay to zero. For any initial value $W_j(0)$, $W_j(t) \to c_j$ 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 $R_k \Sigma_k$. In other words, CGHA converges.

ACKNOWLEDGMENT

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

REFERENCES


