Joint Approximate Diagonalization Using Bilateral Rank-Reducing Householder Transform with Application in Blind Source Separation*

ZHANG Weitao, LIU Ning and LOU Shuntian
(The School of Electronic Engineering, Xidian University, Xi’an 710071, China)

Abstract — This paper addresses the problem of Joint approximate diagonalization (JAD) of a set of given matrices and proposes a new efficient iterative algorithm for JAD that based on the rank-reducing structure of Householder transform. The proposed algorithm, named as HJD, completes the simultaneous diagonalization of the target matrices by successive Householder transform from the point of view of matrix power concentration. Generally, the power of the elements below diagonal element was concentrated to the diagonal element by the rank-reducing Householder transform. Such a particular structure of Householder transform at each iteration prevents the divergence of matrix power. The diagonalization matrix was calculated by the product of all Householder matrices. By applying our algorithm to blind source separation, we demonstrate the efficiency and improvement of the proposed algorithm in estimating the separation matrix.

Key words — Joint approximate diagonalization, Blind source separation (BSS), Householder transform.

I. Introduction

Joint approximate diagonalization of a set of given matrices is a significant signal processing technique in common problems such as blind beamforming\textsuperscript{[1]} and Blind source separation (BSS)\textsuperscript{[2–4]}. Given a target matrix set $\mathcal{R} = \{R_1, R_2, \cdots, R_K\}$, where $R_k \in \mathbb{R}^{N \times N}$, $1 \leq k \leq K$, the joint diagonalization problem seeks a diagonalization matrix $U \in \mathbb{R}^{N \times N}$ such that $\forall k \in \{1, \cdots, K\}$, $UR_kU^T$ is as diagonal as possible. Generally, $R_k$ is generated as follows:

$$R_k = AA_kA_k^T + E_k, \quad \forall k \in \{1, \cdots, K\} \quad (1)$$

where $A_k$ is a real diagonal matrix, $A$ and $E_k$ represent the unknown mixing matrix and the perturbation matrix respectively. The degree of the deviation of the matrix $UR_kU^T$ from diagonality is evaluated by some criterion (cost function). The common criteria include the Weighted least-squares (WLS) criteria of Cardoso used in Refs.\textsuperscript{[1, 2, 4–6]}, the WLS criteria used in Refs.\textsuperscript{[7–9]}, the information theoretic criteria used in Ref.\textsuperscript{[10]}, and the non-WLS criterion used in Ref.\textsuperscript{[11]}.

The joint approximate diagonalization of a set of fourth-order cumulant eigenmatrices (termed as JADE for short) is proposed in Ref.\textsuperscript{[1]}, and later developed in Ref.\textsuperscript{[2]} that used second-order statistics. Both of them consist of “whitening” the observations and followed by “joint diagonalization” of the whole set of statistics matrices with orthogonal constraint on diagonalization matrix $U$. In recent years, a new method for BSS without pre-whitening is developed in Ref.\textsuperscript{[3]}, and a family of nonorthogonal joint diagonalization algorithms that do not impose orthogonal constraint on diagonalization matrix have received increasing attention\textsuperscript{[6–8, 10, 11]}. As a matter of fact, the nonorthogonal joint diagonalization algorithms are generalized version of joint diagonalization. It is readily to see that if the mixing matrix $A$ is orthogonal in Eq.(1), the nonorthogonal joint diagonalization algorithms will converge to an orthogonal diagonalization matrix.

The nonorthogonal joint diagonalization algorithms generally do not need pre-whitening for BSS, they can be applied to the matrix set and estimate mixing matrix or its Moore-Penrose inverse directly. However, the performance of these algorithms will degrade when there exist some perturbations for diagonal matrix $A_k$ in Eq.(1), i.e.,

$$R_k = A(A_k + \sigma_1\xi_k)A_k^T + \sigma_2E_k, \quad \forall k \in \{1, \cdots, K\} \quad (2)$$

where $\xi_k$ is a small matrix with Frobenius norm $\|\xi_k\|_F \rightarrow 0$, $\sigma_1$ and $\sigma_2$ denote the noise level. In practice Eq.(2) is more general for BSS because of the finite sample size of the observations. In this paper, we develop a new efficient joint diagonalization algorithm based on bilateral rank-reducing Householder transform, which is expected to improve the performance of BSS.

The paper is organized as follows. In Section II, we briefly describe the joint diagonalization and BSS problem with the necessary assumptions regarding the source signals and the mixing matrix. In Section III, we derive the HJD algorithm. To demonstrate the efficiency of the HJD algorithm, three different types of simulation results are presented in Section IV. Finally, the paper is concluded in Section V.

*Manuscript Received Feb. 2008; Accepted Feb. 2009. This work is supported by the National Natural Science Foundation of China (No.60775013).
II. Problem Statement

1. Blind separation model

Signals emitted from different sources are observed thanks to

$$x(t) = As(t) + n(t)$$  \hspace{1cm} (3)$$

where $x(t) = [x_1(t), \ldots, x_N(t)]^T$ is the vector of observed signals, vector $s(t) = [s_1(t), \ldots, s_N(t)]^T$ contains the signals emitted by $N$ narrow-band sources, $A$ represents the $N \times N$ square mixing matrix, $n(t)$ is the vector of additive noise. For clarity, we restrict our attention to the case of real signals and mixtures. The common assumptions for BSS are given as follows: (1) Mixing matrix $A$ is time-invariant and has full rank; (2) Source signals $s_i(t)$, $i = 1, \ldots, N$ are zero-mean, unit variance, temporally correlated and mutually independent; (3) The additive noises in Eq.(3) are zero-mean, white Gaussian, variance, temporally correlated and mutually independent.

The first step of JADE and SOBI consists of “whitening” the observed vectors with a whitening matrix $W \in \mathbb{R}^{N \times N}$ such that $WA$ becomes an orthogonal matrix. The second step consists of estimating an orthogonal matrix $U$ by joint diagonalization of the whole set of target matrices (cumulant matrices for JADE and covariance matrices for SOBI). The demixing matrix becomes $B = A^{-1} = UW$. It is readily to see that if $A$ is orthogonal, then the observation $x(t)$ becomes white vector automatically for noise free case, i.e., its covariance matrix $R_x = E[xx^T]$ equals the identity and $W = I$.

Generally, after whitening the white additive noises $n(t)$ in Eq.(3) become color noises. In this paper, we focus our attention on the estimation of orthogonal matrix $U$ (the whitening matrix $W$ can be calculated either by the standard PCA or by an improved whitening method that introduced in Ref.[9]). For the seek of simplicity, we assume that $A$ is orthogonal in the sequel.

2. Joint approximate diagonalization for BSS

The matrix set $\mathcal{R}$ can be generated by various statistics, such as fourth-order cumulant,[1] covariance at different time lags,[2] time-frequency distributions[3] and possibly others. However, they always share the common structure illustrated in Eq.(2). Take the covariance for example, let $\mathcal{R} = \{R_k\} \triangleq \{R_k(\tau_k)\}$ be a given set of covariance matrices, where $\tau_k$ denotes the $k$th time lag. From Eq.(3) and the above assumptions, the covariance matrices have the following form:

$$R_k = E[x(t)x(t + \tau_k)^T] = AR_k(\tau_k)A^T + R_n(\tau_k),$$

$$\forall k \in \{1, \ldots, K\}$$  \hspace{1cm} (4)$$

where $R_k(\tau_k)$ is a diagonal matrix with nonzero diagonal elements according to assumption 2, and $R_n(\tau_k)$ corresponds to the perturbation matrix $E_k$ in Eq.(2).

Let $\mathcal{R} = \{R_k\}$ denotes the estimated matrix set obtained from a finite batch of samples $\{x(t)\}$. Our problem can be described as follows. Given $\mathcal{R}$, estimate the orthogonal diagonalization matrix $U$ such that the following cost function is minimized.

$$J(U) = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} \left| |U \hat{R}_k U^T|_{ij}\right|^2$$  \hspace{1cm} (5)$$

where $[X]_{ij}$ denotes the $ij$th element of matrix $X$. According to Ref.[2], the resulting matrix $U$ is essentially equal to $A$, i.e., $UA = PD$, where $P$ and $D$ are a permutation matrix and a nonsingular diagonal matrix, which represent the well known scale and order indeterminacies in BSS respectively.

III. The Proposed Algorithm

1. Power concentration and Householder matrix

Firstly, the notion of matrix power concentration is introduced. The power of a matrix $M \in \mathbb{R}^{N \times N}$ is defined as its squared Frobenius norm. A useful property of orthogonal transform is that it keeps the norm for any matrix and vector, i.e., $\|M\|_F = \|M\|_F$, where $M = UM$ with $U$ an arbitrary orthogonal matrix. Reconsider Eq.(5) from the point of view of matrix power concentration, we make such conclusion: jointly diagonalizing a target matrix set $\mathcal{R}$ is equivalent to concentrating the power of each matrix to its diagonal elements simultaneously. Now, we solve this problem utilizing Householder transform.

Householder matrix $H \in \mathbb{R}^{N \times N}$ is given as follows[12,13]:

$$H = I - 2vv^T$$  \hspace{1cm} (6)$$

$$V = (v^Tv)^{-1}vv^T$$  \hspace{1cm} (7)$$

where $v = [v_1, v_2, \ldots, v_N]^T$ is a Householder vector. It is readily to see that $HH^T = I$, which guarantees that successive Householder transform makes an orthogonal diagonalization matrix.

2. Algorithm derivation

Without loss of generality, aim at the $i$th ($i \in \{1, \ldots, N - 1\}$) lower triangular column (i.e., the diagonal element and its lower part, l.t.c. for short) of all target matrices $\hat{R}_k$ ($k = 1, \ldots, K$), a Householder matrix $H_i$ is expected to concentrate the power of the $i$th l.t.c. to the $i$th diagonal element. Now, suppose $H_i$ is able to complete this task. Let $\hat{R}_k = H_i \hat{R}_k H_i^{-1}$ denotes the $i$th update of the $k$th matrix in $\mathcal{R}$. So the diagonal element of $\hat{R}_k$ contains the power of the $i$th l.t.c. of $\hat{R}_k^{i-1}$. However, a problem will arise when we continue with the iteration for the $(i + 1)$th l.t.c. of $\hat{R}_k$, because the current update will destroy the previous power concentration results. Herein, we solve this problem by reducing the rank of Householder vector $v$ associated with $H_{i+1}$, i.e., let $v_1 = v_2 = \cdots = v_i = 0$ when computing $H_{i+1}$ from Eqs.(6) and (7). Fig.1 shows the structure of $H_{i+1}$, which guarantees that the first $i$ diagonal elements of $\hat{R}_k^i$ remain unchanged when $\hat{R}_k^i$ is updated by $H_{i+1}$.

The above discussions introduce the power concentration in lower triangular part of the target matrices. Let $\tilde{R}_k = U_k \hat{R}_k$ ($k = 1, \ldots, K$) denotes $N - 1$ update results of the target matrices, where $U_L = \prod_{i=1}^{N-1} H_{N-i}$ is the product of Householder matrices $H_{i+1}$.

![Fig. 1. The rank reduction for vector $v$ and the resulting structure of Householder matrix $H_{i+1}$](image-url)
of $N-1$ Householder matrices used in the previous paragraph. Correspondingly, the upper triangular part of power concentration can be completed as follows. Make transpose of matrices $\hat{R}_k$ $(k = 1, \cdots, K)$, and aim at the lower triangular part of $\hat{R}_k^T$ $(k = 1, \cdots, K)$, calculate $U_R$ using the same procedure as that of $U_L$. Now we get updated target matrices after both lower and upper part power concentration as follows:

$$\hat{R}_k'' = U_R \hat{R}_k^T U_R^T, \quad k = 1, \cdots, K$$  (8)

Make matrices transpose once more, we get a new matrix set

$$\hat{R} = \{\hat{R}_1, \hat{R}_2, \cdots, \hat{R}_K\} = \{U_L \hat{R}_1 U_L^T, U_L \hat{R}_2 U_L^T, \cdots, U_L \hat{R}_K U_L^T\}$$  (9)

and thus complete one full iteration of the proposed algorithm.

It is readily to see that power concentration for the transposed target matrices by left Householder transform is equivalent to the power concentration by right Householder transform.

Repeating the power concentration several times, $\hat{R} = \{U_L \hat{R}_1 U_L^T, U_L \hat{R}_2 U_L^T, \cdots, U_L \hat{R}_K U_L^T\}$ will become a diagonal matrix set, where

$$U_L = \prod_{i=1}^{m} U_L^{(m+i)}$$  (10)

$$U_R = \prod_{i=1}^{m} U_R^{(m+i)}$$  (11)

are the left and right diagonalization matrices respectively, $U_L^{(m+i)}$ and $U_R^{(m+i)}$ denote $U_L$ and $U_R$ obtained in the $(m+i)$th iteration, $m$ represents the number of full iterations. For the left and right diagonalization matrices, we have the following theorem.

**Theorem 1** Given a matrix set $\hat{R} = \{R_1, R_2, \cdots, R_K\}$, assume, $\forall k \in \{1, \cdots, K\}$, $R_k$ is symmetrical and has distinct squared eigenvalues. If there exist $U$ and $V$ being the left and right orthogonal diagonalization matrices such that $\forall k \in \{1, \cdots, K\}$, $D_k = U R_k V^T$ is a diagonal matrix, then both the left and right diagonalization matrices can be regarded as the diagonalization matrix, and they are related by the following equation.

$$U = JV$$  (12)

where $J = diag[\pm 1]$ is a diagonal matrix with the diagonal elements equal to 1 or $-1$.

The assumption for theorem 1 is generally satisfied in the BSS context. The proof of this theorem is presented in Appendix A. To prove theorem 1, we need to present the following lemma.

**Lemma 1** If a matrix $R \in \mathbb{R}^{N \times N}$ has $N$ distinct eigenvalues $\lambda_i$, $i = 1, \cdots, N$, then two eigenvectors associated with the same eigenvalue are linear related, i.e.,

$$x_i = \mu y_i$$  (13)

where $\mu \neq 0$ is an arbitrary constant, $x_i$ and $y_i$ are two eigenvectors associated with $\lambda_i$.

**Proof** See Appendix A.

The only remaining problem is to calculate the Householder matrix $H$ for each power concentration. Without loss of generality, consider the $i$th l.t.c. $(i \in \{1, \cdots, N-1\})$ of all target matrices $\hat{R}_k$ $(k = 1, \cdots, K)$, which are selected to form the matrix $G_i \in \mathbb{R}^{(N-i+1) \times K}$, i.e., the $k$th column of $G_i$ consists of $i$th l.t.c. of the $k$th target matrix $\hat{R}_k$. We now rewrite the Householder matrix $H_i$ as $H_i = [h_{1i}, h_{2i}, \cdots, h_{N-i+1i}]^T$, where $\forall i \in \{1, \cdots, N-i+1\}$, $h_{ki}$ denotes the $i$th row of $H_i$.

Since the Householder matrix $H_i$ is orthogonal, the product of $H_i$ and $G_i$ keeps the overall power of $G_i$ unchanged. So minimizing Eq.(5) is equivalent to maximizing the squared sum of diagonal elements of target matrices. The matrix power concentration problem is converted to find $H_i$ to maximize the squared norm of $f_1$ by assuming

$$f_1 = H_i G_i = \left(\begin{array}{c} h_{11} \\ h_{21} \\ \vdots \\ h_{N-i+1,1} \end{array}\right) G_i = \left(\begin{array}{c} h_{11} G_{i1} \\ h_{21} G_{i2} \\ \vdots \\ h_{N-i+1,1} G_{i(N-i+1)} \end{array}\right) = \left(\begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_{N-i+1} \end{array}\right)$$  (14)

where $f_1 = h_{1i} G_i$ represents the first row of $f_i$, the squared norm of $f_1$ can be written as

$$\|f_1\|^2 = f_1^T f_1 = h_{1i} (G_i G_i^T) h_{1i}^T$$  (15)

Maximizing a quadratic form under the unit norm constraint of its argument is typically obtained by taking to be the eigenvector of $G_i G_i^T$ associated with the largest eigenvalue. Let $h_1 = [h_{11}, h_{12}, \cdots, h_{1(N-i+1)}]$, contrast $h_1$ with the first row of $H_i$ that explicitly determined by $v$, we have the following equations:

$$2v_i^2 = 1 - h_{11}, \quad 2v_i v_{i+1} = -h_{12}, \quad \cdots, \quad 2v_i v_N = -h_{1(N-i+1)}$$  (16)

where $\|v\|^2 = \sum_{n=1}^{N} v_n^2$. In practice, it is not necessary to solve Eq.(16) because our objective is to calculate $V$ in Eq.(7). Let $a = 2v_i/\|v\|^2$, $\tilde{v} = [v_1, v_1+1, \cdots, v_N]^T$, and

$$\tilde{a} = [1 - h_{11}, -h_{12}, \cdots, -h_{1(N-i+1)}]^T$$  (17)

Eq.(16) can be written in a vector form

$$a \tilde{v} = \tilde{a}$$  (18)

Substitution of Eq.(18) into Eq.(7) yields

$$V = vv^T/v^Tv = aa^T/a^Ta$$  (19)

where $v = [0, 0, \cdots, 0, v_i^T]^T$, and $a = [0, 0, \cdots, 0, \tilde{a}^T]^T$. Now we summarize HJD algorithm as follows:

- **Input**: The target matrix set $\hat{R} = \{\hat{R}_1, \hat{R}_2, \cdots, \hat{R}_K\}$.
- **Output**: Orthogonal diagonalization matrix $U$.
- **Initialize** $U_L, U_R$ with an identity matrix.
- **Implement** the following pseudo code till convergence or a termination condition is met.

  For $m = 1, \cdots, M$ do
  (1) Initialize $U_L, U_R$ with an identity matrix.
  (2) For $i = 1, \cdots, N-1$ do
        (i) Select the $i$th l.t.c. of all target matrices $\hat{R}_k (k = 1, \cdots, K)$ to form $G_i$.
        (ii) Let $h_1$ equal the eigenvector of $G_i G_i^T$ associated with the largest eigenvalue.
        (iii) Calculate $V$ from Eq.(16)–Eq.(19), followed by Eq.(6) to compute $H_i$.
        (iv) Update all target matrices by $H_i$, let $U_L = H_i U_L$.
    End (For $i$)
  (3) Make transpose of all updated target matrices. Repeat step 2, and exchange $U_L$ and $U_R$.
  (4) Let $U_L = U_L U_L^T$ and $U_R = U_R U_R^T$, make transpose of all updated target matrices again.
  End (For $m$)
\( U = U_L \) or \( U = U_R \).

Note that various termination conditions can be used to stop the iterations of HJD. For example, we may stop the iterations at the \((m+1)\)th iteration once the following condition is satisfied
\[
|J(U^{(m+1)}) - J(U^{(m)})| \leq \varepsilon
\]
where \( \varepsilon \) is a small positive constant and \( U^{(m)} \) is the diagonalization matrix obtained at the \(m\)th iteration.

IV. Computer Simulation Results

In this section, we investigate the performance of HJD algorithm, and compare it with other three joint diagonalization algorithms: ARD\[^6\], ACDC\[^7\], and FAJ\[^11\]. For the sake of simplicity, we constrain the mixing matrix \( B \) for BSS since the whitening matrix becomes identity.

Convergence algorithms will converge to an orthogonal diagonalization \( \xi \) with noise level \( \varepsilon \). The convergence curves of 100 independent runs for four algorithms are initialized with identity matrix.

In order to measure the performance of joint diagonalization algorithms, we use the performance index defined by \[^4\]
\[
PI = \sum_{i=1}^{N} \left( \frac{\sum_{j=1}^{N} |c_{ij}|}{\max_k |c_{ik}|} - 1 \right) + \sum_{j=1}^{N} \left( \frac{\sum_{i} |c_{ij}|}{\max_k |c_{ik}|} - 1 \right)
\]
where \( C = \{c_{ij}\} = BA = UA \) is the global mixing-demixing matrix and \( \max_k |c_{ik}| \) represents the maximal value among the elements in the \(i\)th row of \( C \).

Example 1 In this example, we investigate the typical convergence patterns of cost function \( J(U) \) under the case of Eq.(2). All the elements in \( A_k \), \( \xi_k \), \( E_k \) and the diagonal elements in \( A_k \) are generated independently from the standard normal distribution, and \( A_k \) is orthonormalized by some technique to satisfy the experimental condition. Let \( \xi_k = \xi_k \xi_k^T \) and \( E_k = E_k E_k^T \) to make \( R_k \) symmetrical. The four algorithms are initialized with identity matrix.

Choosing \( N = 5 \), \( K = 10 \), Fig.2(a) plots average convergence curves of 100 independent runs for four algorithms with noise level \( \sigma_1 = 0 \), \( \sigma_2 = 0.1 \). We can see that FAJ algorithm shows fastest convergence. However, the diagonalization error is intolerable. HJD algorithm shows faster convergence than ACDC and ARD, but the diagonalization error is slightly large. In Fig.2(b), we demonstrate the similar curves with noise level \( \sigma_1 = 0.05 \), \( \sigma_2 = 0.1 \). It is readily to see that HJD algorithm provides better performance under the case of disturbance for diagonal matrix \( A_k \).

Example 2 In this example, four algorithms are applied to BSS using JADE. The five source signals are given by

\[
S(t) = \begin{bmatrix}
\begin{array}{l}
s_1(t) \\
s_2(t) \\
s_3(t) \\
s_4(t) \\
s_5(t)
\end{array}
\end{bmatrix} = \begin{bmatrix}
\cos(2\pi 100t^2) \\
\sin(2\pi 25t) \sin(2\pi 800t) \\
\sin(2\pi 300t + 6 \cos(2\pi 60t)) \\
AR(p) \\
ARMA(p,q)
\end{bmatrix}
\]

where \( AR(p) \) and \( ARMA(p,q) \) are the order \( p \)-order \( AR \) and \( p,q \)-order \( ARMA \) random processes (here we chose \( p = q = 2 \)) driven by two independent white noises with zero mean uniform distributions for each independent runs. The \( AR \) model is given by
\[
T_{AR}(z) = \frac{1}{(1 + 0.75z^{-1})(1 + 0.9z^{-1})}
\]
and the \( ARMA \) model is
\[
T_{ARMA}(z) = \frac{(1 + (0.5 + 0.4i)z^{-1})(1 + (0.5 - 0.4i)z^{-1})}{(1 + 0.7z^{-1})(1 + 0.8z^{-1})}
\]

We would like to point out that the five source signals used here in Example 3 are rescaled to satisfy the requirement of unit variance. The mixing matrix is randomly generated as

![Fig. 3. Average performance of four algorithms for BSS using JADE with noise level std=0.1](image)

![Fig. 4. Average performance of four algorithms for BSS using SOBI. (a) Average performance versus sample size with noise level std=0.1; (b) average performance versus noise level](image)
Example 1, and the observations are obtained from Eq. (3). The noises \( n(t) \) are normally distributed with standard deviation (std) 0.1. Choosing \( K = 15 \) for each sample size, we calculate 15 cumulant matrices to form the target matrix set \( \mathbf{R} \). Fig.3 plots the average PI of 100 independent runs for each sample size.

Example 3 In this example, we compare the performance of four algorithms by using SOBI. The mixing matrix, source signals, and additive noises are the same as that in Example 2. The cumulant matrices are substituted for covariance matrices at different time lags to form the target matrix set \( \mathbf{R} \). Choosing \( K = 15 \) we demonstrate the average performance of four algorithms over 200 independent runs in Fig. 4.

Fig.4(a) plots the average PI versus the number of samples for four competitors with noise level 0.1. In Fig.4(b), the average PI is plotted as a function of noise level (which is expressed in decibels) with 5000 samples for each independent run. The results show that HJD algorithm exhibits better performance.

V. Conclusion

In this paper, we develop a new joint diagonalization algorithm based on the bilateral rank-reducing Householder transform, which makes full use of the property of orthogonal transform. From the point of view of matrix power concentration, we exploit a rank-reducing structure of Householder matrix in each loop to prevent the divergence of matrix power. Although we get different diagonalization matrices (left and right diagonalization matrices), it is proved that both of them can be regarded as the diagonalization matrix. In the end, computer simulation results are given to show the performance improvement of HJD algorithm in comparison to the existing algorithms for BSS.

Appendix A

Proof of Theorem 1 \( \forall k \in \{1, \cdots, K\} \), let \( \mathbf{D}_k = \mathbf{D}_k^T = \mathbf{D}_k \mathbf{D}_k^T, \mathbf{R}_k = \mathbf{R}_k^T \) and note that \( \mathbf{R}_k = \mathbf{R}_k^T, \mathbf{V}^T \mathbf{V} = \mathbf{I} \) we have

\[
\mathbf{D}_k = \mathbf{U} \mathbf{R}_k \mathbf{U}^T \quad (A.1)
\]

Similarly, let \( \mathbf{D}_k = \mathbf{D}_k^T = \mathbf{D}_k^T \mathbf{D}_k, \mathbf{R}_k = \mathbf{R}_k^T \mathbf{V}^T \mathbf{V} = \mathbf{I} \) we get

\[
\mathbf{D}_k = \mathbf{V} \mathbf{R}_k \mathbf{V}^T \quad (A.2)
\]

Let \( \mathbf{D}_k = \text{diag}(d_1, \cdots, d_N) \), where \( d_i \) denotes the ith diagonal element of \( \mathbf{D}_k \). Let \( \mathbf{U} = [u_1^T, u_2^T, \cdots, u_N^T]^T, \mathbf{V} = [v_1^T, v_2^T, \cdots, v_N^T]^T \), where \( u_i \) and \( v_i \) represent the ith row of \( \mathbf{U} \) and \( \mathbf{V} \). Postmultiplying Eq.(A.1) and Eq.(A.2) by \( \mathbf{U} \) and \( \mathbf{V} \) respectively, we have

\[
u_i \mathbf{R}_k = d_i u_i, \quad i = 1, \cdots, N \quad (A.3)
\]

\[
u_i \mathbf{R}_k = d_i v_i, \quad i = 1, \cdots, N \quad (A.4)
\]

It is readily to see that \( u_i \) and \( v_i \) are two eigenvectors of \( \mathbf{R}_k \) associated with the same eigenvalue \( d_i \). Combined with the assumption \( \forall 1 \leq i \neq j \leq N, d_i \neq d_j \) and using Lemma 1, we have

\[
u_i = \mu v_i, \quad i = 1, \cdots, N \quad (A.5)
\]

where \( \mu \neq 0 \) is an arbitrary constant. Note that \( \mathbf{U} \) and \( \mathbf{V} \) are orthogonal, i.e., \( \|u_i\| = \|v_i\| = 1 \) one immediately get \( \mu = \pm 1 \), followed by

\[
\mathbf{U} = [u_1^T, u_2^T, \cdots, u_N^T]^T = [\pm v_1^T, \pm v_2^T, \cdots, \pm v_N^T]^T
\]

\[
\mathbf{J} = \text{diag}(\pm 1)
\]

where \( \mathbf{J} = \text{diag}(\pm 1) \).

\[
\forall k \in \{1, \cdots, K\}, \text{consider Eq. (A.6), we have} \quad \mathbf{J}\mathbf{D}_k = \mathbf{J}\mathbf{U}\mathbf{R}_k \mathbf{V}^T = \mathbf{V} \mathbf{R}_k \mathbf{V}^T \mathbf{J} \text{ and } \mathbf{D}_k \mathbf{J} = \mathbf{U}\mathbf{R}_k \mathbf{V}^T \mathbf{J} = \mathbf{U}\mathbf{R}_k \mathbf{U}^T \mathbf{J}, \text{noticing that } \mathbf{J}\mathbf{D}_k \text{ and } \mathbf{D}_k \mathbf{J} \text{ are still diagonal, i.e., both } \mathbf{U} \text{ and } \mathbf{V} \text{ can diagonalize the matrix set } \mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_K\} \text{. Therefore, both } \mathbf{U} \text{ and } \mathbf{V} \text{ can be used as the diagonalization matrix. This completes the proof of Theorem 1.}

Proof of Lemma 1 According to the definition of eigenvalue and eigenvector, we immediately have

\[
(R - \lambda_i \mathbf{I}) \mathbf{x}_i = 0, \quad i \in \{1, \cdots, N\} \quad (A.7)
\]

Similarly, we have

\[
(R - \lambda_i \mathbf{I}) \mathbf{y}_i = 0, \quad i \in \{1, \cdots, N\} \quad (A.8)
\]

From Eq.(A.7) and Eq.(A.8), we get \( \mathbf{x}_i \in \text{Null}(R - \lambda_i \mathbf{I}); \mathbf{y}_i \in \text{Null}(R - \lambda_i \mathbf{I}) \), where \( \text{Null}(\cdot) \) represents the null space of a matrix. It is straightforward that if \( \text{dim}(\text{Null}(R - \lambda_i \mathbf{I})) = 1 \), then \( \mathbf{x}_i \) and \( \mathbf{y}_i \) are linear related, i.e., \( \mathbf{x}_i = \gamma \mathbf{y}_i \). To this end, let \( R = Q \text{diag}(\lambda_1, \cdots, \lambda_N) \mathbf{Q}^T \) denote the eigenvalue decomposition of \( \mathbf{R} \), and considering \( \forall i \neq j, \lambda_i \neq \lambda_j \), we have

\[
\text{rank}(R - \lambda_i \mathbf{I}) = \text{rank}(Q \text{diag}(\lambda_1 - \lambda_i, \cdots, \lambda_N - \lambda_i)) \mathbf{Q}^T = \text{rank}(\text{diag}(\lambda_1 - \lambda_i, \cdots, \lambda_N - \lambda_i)) = N - 1 \quad (A.9)
\]

Note that\(^{[13]}\)

\[
\text{rank}(R - \lambda_i \mathbf{I}) + \text{dim}(\text{Null}(R - \lambda_i \mathbf{I})) = N \quad (A.10)
\]

one immediately has

\[
\text{dim}(\text{Null}(R - \lambda_i \mathbf{I})) = 1 \quad (A.11)
\]

This completes the proof of Lemma 1.

References


ZHANG Weitao was born in Shaanxi Province, China. He received the B.S. degree in electrical engineering from Xidian University, Xi'an, China, in 2006. He is currently working toward the Ph.D. degree at the School of Electronic Engineering, Xidian University. His research interests include blind signal processing and communication signal processing in MIMO-OFDM system.

LIU Ning received the B.S. degree in electrical engineering from Xidian University, Xi'an, China, in 2004 and is currently working toward the Ph.D. degree at Xidian University. His current interests focus on blind source separation and image restoration.

LOU Shuntian was born in Zhejiang Province, China, in 1962. He received the B.S. degree in automatic control and M.S. degree in electronic engineering from Xidian University, Xi'an, China, and the Ph.D. degree in navigation guidance and control from Northwest Polytechnical University, Xi'an, China, in 1985, 1988, and 1999, respectively. From 1999 to 2002, he was a Postdoctoral Fellow at the Institute of Electronic Engineering, Xidian University. Currently, he is a Professor at the School of Electronic Engineering, Xidian University. His research interests are in the areas of signal processing, pattern recognition, and intelligent control using neural network and/or fuzzy system.