RECURSIVE COMPLEX BLIND SOURCE SEPARATION VIA EIGENDECOMPOSITION OF CUMULANT MATRICES

Puskal P. Pokharel[1], Umut Ozertem[2], Deniz Erdogmus[2], Jose C. Principe[1]

[1] Computational NeuroEngineering Laboratory, ECE Department, University of Florida
[2] Adaptive Systems Laboratory, CSEE Department, Oregon Health and Science University

ABSTRACT

Under the assumptions of non-Gaussian, non-stationary, or non-white independent sources, linear blind source separation can be formulated as a generalized eigenvalue decomposition problem. Here we provide an elegant method of doing this online, instead of waiting for a sufficiently large batch of data. This is done through a recursive generalized eigendecomposition algorithm that tracks the optimal solution, which is obtained using all the data observed. The algorithms proposed in this paper follow the well-known recursive least squares (RLS) algorithm in nature.

Index Terms— Independent component analysis, blind source separation, generalized eigendecomposition, cumulants

1. INTRODUCTION

Independent component analysis (ICA) is an important statistical tool in signal processing and machine learning, both as a solution to the problem of blind source separation (BSS) [1,2] and as a preprocessing step that complements a more comprehensive solution as in dimensionality reduction and feature extraction [3,4]. To implement these applications feasibly on contemporary digital signal processors (DSP), online learning algorithms are required.

Currently, the online ICA solutions are obtained using algorithms designed with the stochastic gradient concept (e.g., Infomax [6]). The drawbacks of stochastic gradient algorithms in online leaning include difficulty in selecting the step size for optimal speed misadjustment trade off and suboptimal estimates of the weights given all the samples seen at any given iteration.

Recursive least squares (RLS) is an online algorithm for supervised adaptive filter training, which has the desirable property that the estimated weights correspond to the optimal least squares solution that one would obtain using all the data observed so far, provided that initialization is done *properly* [7]. This benefit comes at a cost of additional computational requirements compared to LMS. Nevertheless, it would be beneficial in certain ICA applications to track at each step the optimal solution given all the data up to the step. The joint diagonalization of higher order statistics have been known to solve the ICA problem under the assumed linear mixing model and have lead to popular algorithms like JADE [8]. This motivates the derivation of a recursive generalized eigendecomposition (GED) based ICA algorithm that is similar to RLS in principle, but solved by the simultaneous diagonalization of the second and fourth order joint statistics of the observed mixtures. This can be done in three major ways, assuming the sources are non-stationary and decorrelated [9], non-white and decorrelated [10], or nonGaussian and independent [11].

In this paper we contribute recursive BSS algorithms (RBSS) based on recursive generalized eigendecompositions of cumulants. The algorithms are demonstrated on separation of instantaneous linear mixtures of speech through Monte Carlo simulations.

2. RECURSIVE ICA ALGORITHM

The square linear ICA problem can be expressed as

$$\mathbf{X} = \mathbf{AS} \tag{1}$$

where $\mathbf{X}$ is the $n \times N$ observations matrix, $\mathbf{A}$ is the $n \times n$ mixing matrix, and $\mathbf{S}$ is the $n \times N$ independent sources matrix. If we consider each column as a sample in time, (1) becomes

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t \,. \tag{2}$$

The joint diagonalization of the covariance matrices and higher order cumulant matrices can be compactly formulated in the form of a generalized eigendecomposition problem that gives the ICA solution an analytical form [12]. According to this formulation, under the assumptions of non-Gaussian, non-stationary, or non-white sources, the separation matrix $\mathbf{W}$ is the solution to the following generalized eigenvalue problem,

$$\mathbf{R_X W} = \mathbf{Q_X W}\Lambda \,, \tag{3}$$

where $\mathbf{R}_x$ is the covariance matrix and $\mathbf{Q_X}$ is a cumulant matrix for non-Gaussian and independent sources, the covariance matrix at a different time instant for non-stationary and decorrelated sources, or the cross-correlation matrix for a certain time delay for non-white and decorrelated sources. Λ is a diagonal matrix related to the cross statistics of the sources. Here the recursive algorithms for these cases are presented.

2.1. Non-Gaussian and independent sources

For this case, $\mathbf{Q_X}$ is the cumulant matrix estimated using sample averages. While any order of cumulants can be employed, lower orders are more robust to outliers and small sample sizes, so we focus on the fourth order cumulant matrix, which is given as

$$\mathbf{Q_X} = E\left[\mathbf{x}^H\mathbf{x}\mathbf{x}\mathbf{x}^H\right] - \mathbf{R_X}\mathit{trace}(\mathbf{R_X}) - E\left[\mathbf{x}\mathbf{x}^T\right]E\left[\mathbf{x}^*\mathbf{x}^H\right] - \mathbf{R_X}\mathbf{R_X} \tag{4}$$

where $\mathbf{x}^H$, $\mathbf{x}^T$ and $\mathbf{x}^*$ represent Hermitian transpose, transpose, and complex conjugate of $\mathbf{x}$, respectively. With independent identically distributed samples, expectations reduce to sample averages for covariance and cumulant matrices.

Then one can define recursive update rules for the estimates of the covariance and cumulant matrices, $\mathbf{R}$ and $\mathbf{Q}$. The recursive update rule for the covariance matrix is

$$\mathbf{R}_t = \frac{t-1}{t}\mathbf{R}_{t-1} + \frac{1}{t}\mathbf{x}_t\mathbf{x}_t^H \qquad (5)$$

and the update rule for the cumulant matrix is given by

$$\mathbf{Q}_t = \mathbf{C}_t - \mathbf{B}_t\mathbf{B}_t^* - \mathbf{R}_x trace(\mathbf{R}_x) - \mathbf{R}_x^2 . \qquad (6)$$

$\mathbf{C}$ is $E[\mathbf{x}^H\mathbf{x}\mathbf{x}\mathbf{x}^H]$ and its estimate can be updated as

$$\mathbf{C}_t = \frac{t-1}{t}\mathbf{C}_{t-1} + \frac{1}{t}(\mathbf{x}_t^H\mathbf{x}_t)\mathbf{x}_t\mathbf{x}^H . \qquad (7)$$

$\mathbf{B}$ is $E[\mathbf{x}\mathbf{x}^T]$ and its estimate can be updated as

$$\mathbf{B}_t = \frac{t-1}{t}\mathbf{B}_{t-1} + \frac{1}{t}\mathbf{x}_t\mathbf{x}_t^T . \qquad (8)$$

The following recursive update of $\mathbf{R}^2$ can be obtained by squaring (5).

$$\mathbf{R}^2_t = \frac{(t-1)^2}{t^2}\mathbf{R}^2_{t-1} + \frac{1}{t^2}(\mathbf{x}_t^H\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^H + \frac{t-1}{t^2}[\mathbf{v}_t\mathbf{x}_t^H + \mathbf{x}_t\mathbf{v}_t^H] \qquad (9)$$

For further computational savings we introduce the vector $\mathbf{v}_t$ as $\mathbf{v}_t = \mathbf{R}_{t-1}\mathbf{x}_t$ and we can obtain $\mathbf{R}^{-1}$ and $\mathbf{R}^{-1}\mathbf{Q}$ by iterating to avoid matrix multiplications and inversions having $O(n^3)$ computational load. These two matrices are required for the fixed point algorithm that solves for the generalized eigendecomposition which is discussed later. Employing the matrix inversion lemma [7], the recursion rule for $\mathbf{R}^{-1}$ becomes

$$\mathbf{R}_t^{-1} = \frac{t}{1-t}\mathbf{R}_{t-1}^{-1} - \frac{t}{(t-1)\alpha_t}\mathbf{u}_t\mathbf{u}_t^H \qquad (10)$$

where α_t and $\mathbf{u}_t$ are defined as

$$\alpha_t = (t-1) + \mathbf{x}_t^H\mathbf{u}_t \qquad \mathbf{u}_t = \mathbf{R}_{t-1}^{-1}\mathbf{x}_t . \qquad (11)$$

Here we also define the matrix $\mathbf{D}$ as

$$\mathbf{D}_t = \mathbf{R}_t^{-1}\mathbf{Q}_t . \qquad (12)$$

A recursive update rule can also be obtained for $\mathbf{D}$ through combination and simple manipulation of equations (6) and (10) to save computation.

2.2. Non-stationary and decorrelated sources

In this case, $\mathbf{Q}_\mathbf{x}$ is the covariance matrix $E[\mathbf{x}_k\mathbf{x}_k^H]$ where k is at a different time than that used to calculate $\mathbf{R}_x$. For computation, we can estimate the expectations of the two covariance matrices by sample averages of the data points in non-overlapping windows, both with lengths close to the stationarity time of the signals. Then the update rule for $\mathbf{Q}_\mathbf{x}$ is

$$\mathbf{Q}_k = \frac{k-1}{k}\mathbf{Q}_{k-1} + \frac{1}{k}\mathbf{x}_k\mathbf{x}_k^H \qquad (13)$$

and that for $\mathbf{R}_x$ is

$$\mathbf{R}_t = \frac{t-1}{t}\mathbf{R}_{t-1} + \frac{1}{t}\mathbf{x}_t\mathbf{x}_t^H \qquad (14)$$

where $k=1$ and $t=1$ are the beginning times for the two non-overlapping windows on the data.

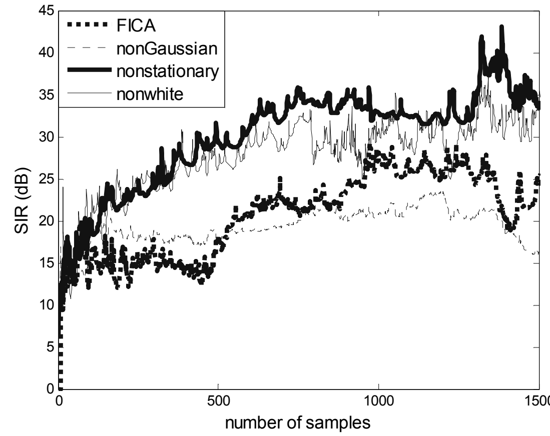

Figure 1. SIR (dB) for Fast-ICA, and the proposed RBSS using the assumptions of nonGaussianity, nonstationarity and nonwhiteness as described in section 2 with the condition number of the mixing matrix as 40.

2.3. Non-white and decorrelated sources

When the sources are non-white and decorrelated, $\mathbf{Q}_\mathbf{x}$ in (3) can be taken as the symmetric cross-correlation matrix with a time delay, i.e.,

$$\mathbf{Q}_x = E[\mathbf{x}_t\mathbf{x}_{t+\tau}^H + \mathbf{x}_{t+\tau}\mathbf{x}_t^H] . \qquad (15)$$

This can be estimated online using

$$\mathbf{Q}_t = \frac{t-1}{t}\mathbf{Q}_{t-1} + \frac{1}{t}(\mathbf{x}_t\mathbf{x}_{t+\tau}^H + \mathbf{x}_{t+\tau}\mathbf{x}_t^H) . \qquad (16)$$

τ is chosen so that autocorrelation terms in $\mathbf{Q}_\mathbf{x}$ are nonzero.

2.4. Deflation procedure

Having the update equations, the aim is to find the optimal solution for the eigendecomposition for the updated correlation and cumulant matrices in each iteration. As given by (3) we need to solve for the weight matrix $\mathbf{W}$. We will employ the deflation procedure to determine each generalized eigenvector sequentially. Every generalized eigenvector $\mathbf{w}_d$ that is a column of $\mathbf{W}$ is a stationary point of the function

$$J(\mathbf{w}) = \frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\mathbf{w}^H\mathbf{Q}\mathbf{w}} . \qquad (17)$$

This fact can easily be observed by taking the derivative of the expression on the right of (17) with respect to $\mathbf{w}$, and equating it to zero which will result in

$$\mathbf{R}\mathbf{w} = \frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\mathbf{w}^H\mathbf{Q}\mathbf{w}}\mathbf{Q}\mathbf{w} . \qquad (18)$$

This is the equation for generalized eigendecomposition, the eigenvalues being the value of the objective function $J(\mathbf{w})$ given in (17) evaluated at its stationary points. Thus the fixed point algorithm becomes

$$\mathbf{w} \leftarrow \frac{\mathbf{w}^H\mathbf{R}\mathbf{w}}{\mathbf{w}^H\mathbf{Q}\mathbf{w}}\mathbf{R}^{-1}\mathbf{Q}\mathbf{w} . \qquad (19)$$

This fixed point optimization procedure converges to the largest

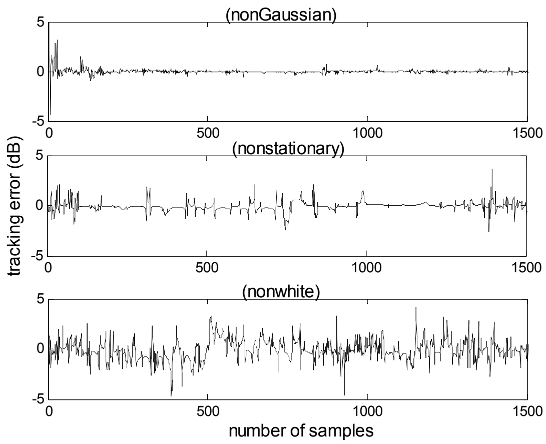

Figure 2. Performance difference between the original GED-BSS and the proposed RBSS for the three different assumptions of nonGaussianity, nonstationarity and nonwhiteness.

generalized eigenvector (the one corresponding to the largest eigenvalue) of $\mathbf{R}$ and $\mathbf{Q}$, and the deflation procedure is employed to manipulate the matrices such that they have the same generalized eigenvalue and eigenvector pairs except for the ones that have been determined previously [13]. The larger eigenvalues are replaced by zeros in each deflation step. Note that in this subsection the time index is implicit and omitted for notational convenience. With d denoting the dimension index, the deflation procedure employed while iterating the dimensions is given by

$$\mathbf{Q}_d = \left[I - \frac{\mathbf{Q}_{d-1}\mathbf{w}_{d-1}\mathbf{w}_{d-1}^H}{\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}\mathbf{w}_{d-1}} \right]\mathbf{Q}_{d-1}$$
$$\mathbf{R}_d = \mathbf{R}_{d-1}. \quad (20)$$

The deflated matrices are initialized to $\mathbf{Q}_1 = \mathbf{Q}$ and $\mathbf{R}_1 = \mathbf{R}$. Obtaining the new matrices, we employ the same fixed point iteration procedure given in (19) to find the corresponding eigenvector. Given (19), it is clear that iterating $\mathbf{R}^{-1}$ and $\mathbf{D}$ as suggested will result in computational savings. The deflation rules for these matrices can be deduced easily. The deflation of $\mathbf{R}^{-1}$ is

$$\mathbf{R}_d^{-1} = \mathbf{R}_{d-1}^{-1}. \quad (21)$$

Similarly, the deflation rule for $\mathbf{D}$ can be obtained by combining (20) and (21) resulting in

$$\mathbf{D}_d = \left[\mathbf{I} - \frac{\mathbf{w}_{d-1}\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}}{\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}\mathbf{w}_{d-1}} \right]\mathbf{D}_{d-1}. \quad (22)$$

For each generalized eigenvector, the corresponding fixed-point update rule then becomes

$$\mathbf{w}_d \leftarrow \frac{\mathbf{w}_d^H\mathbf{R}_d\mathbf{w}_d}{\mathbf{w}_d^H\mathbf{Q}_d\mathbf{w}_d}\mathbf{D}_d\mathbf{w}_d. \quad (23)$$

Employing this fixed-point algorithm for each dimension and solving for the eigenvectors sequentially, one can update $\mathbf{W}$ and proceed to the next time update step. The combination of these weight updates, matrix deflation procedures, and recursive covariance/cumulant updates give us the Recursive BSS (RBSS) algorithms for the three sets of assumptions. The algorithms are summarized in Table 1. In theory, these recursive algorithms are expected to track the batch GED solutions that one would obtain at any given time using all the data available up to that point. In practice, random initialization and numerical errors (in updates and fixed-point iterations) culminate in some deviation.

3. EXPERIMENTS AND RESULTS

We present results comparing the original GED-BSS algorithms [13] with the results of the proposed RBSS algorithm. For reference, FastICA [14] results are included in the comparisons. Although we initially attempted to include comparisons with the stochastic gradient based Infomax [6] algorithm, finding a large stable stepsize for each individual run proved to be challenging, therefore these results are omitted. The experiments include the separation of speech signals from instantaneous linear mixtures. The database consists of 10 clips of acoustic signals (5 male, 4 female, 1 symphony). We select 9 random pairs from this set and run 10 Monte Carlo (MC) simulations for each pair. In each MC run, a mixing matrix with constant condition number is generated, and the RBSS algorithms are randomly initialized to small diagonal correlation matrices and random weight matrices. Original GED-BSS and Fast-ICA algorithms both run on a batch of data, with the batch size increasing by one sample in each iteration. The RBSS algorithms operate on-line updating matrices and weights using one new sample at a time. All RBSS algorithms and FastICA are allowed 5 fixed-point updates per new sample using their respective update rules. This means, Fast ICA implements 5 fixed point iterations over the whole available data set at any given time. Comparisons are provided using the standard average signal-to-interference ratio (SIR) measure in decibels (dB) [15].

Figure 1 shows the performance of Fast ICA, original GED-BSS and the three RBSS methods for mixture condition number of 40. The results were very similar, as we would expect for smaller condition numbers. In these cases, the convergence speed is not affected by the mixture condition number. The two algorithms using non-Gaussianity and independence assumptions, recursive ICA [16] and FastICA perform worse than the RBSS algorithms using the more suitable non-stationarity and non-whiteness assumptions for speech. Figure 2 shows the tracking error between the RBSS algorithms and their corresponding GED-BSS algorithms. The asymptotic tracking error could be made arbitrarily small by letting RBSS algorithms iterate more per sample.

6. CONCLUSION

Online ICA/BSS algorithms are essential for many signal processing and machine learning applications, where the ICA solution acts as a front-end preprocessor, a feature extractor, or a portion of a solution to a larger problem. Though stochastic gradient based algorithms motivated by various ICA criteria is used in such situations with the advantage of yielding computationally simple weight update rules, they do not offer an optimal solution at every iteration and choosing an *appropriate* step size is still an inconvenience. In this paper we presented recursive BSS algorithms based on the joint digitalization of various cross statistics based on three standard assumption sets regarding source signals: non-Gaussianity, non-stationarity, and non-whiteness. The derivation employs the use of the matrix inversion lemma and the update rules for the expectations approximated by sample averages. The resulting algorithm, of course, is computationally more expensive than stochastic gradient type algorithms per update. However, it converges to and tracks

Assumptions on the sources	Cross-statistics ($\mathbf{Q}_x$) used	Recursion rules
non-Gaussian and independent [11, 12]	$E\left[\mathbf{x}^H\mathbf{x}\mathbf{x}\mathbf{x}^H\right]-\mathbf{R}_{\mathbf{X}}trace(\mathbf{R}_{\mathbf{X}})$ $-E\left[\mathbf{x}\mathbf{x}^T\right]E\left[\mathbf{x}^*\mathbf{x}^H\right]-\mathbf{R}_{\mathbf{X}}\mathbf{R}_{\mathbf{X}}$	$\mathbf{R}_t=\frac{t-1}{t}\mathbf{R}_{t-1}+\frac{1}{t}\mathbf{x}_t\mathbf{x}_t{}^H$ $\quad\mathbf{B}_t=\frac{t-1}{t}\mathbf{B}_{t-1}+\frac{1}{t}\mathbf{x}_t\mathbf{x}_t{}^T$ $\mathbf{Q}_t=\mathbf{C}_t-\mathbf{B}_t\mathbf{B}_t^*-\mathbf{R}_x trace(\mathbf{R}_x)-\mathbf{R}_x{}^2$ $\quad\mathbf{R}^2_t=\frac{(t-1)^2}{t^2}\mathbf{R}^2_{t-1}+\frac{1}{t^2}(\mathbf{x}_t{}^H\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t{}^H+\frac{t-1}{t^2}[\mathbf{v}_t\mathbf{x}_t{}^H+\mathbf{x}_t\mathbf{v}_t{}^H]$ $\mathbf{C}_t=\frac{t-1}{t}\mathbf{C}_{t-1}+\frac{1}{t}(\mathbf{x}_t{}^H\mathbf{x}_t)\mathbf{x}_t\mathbf{x}^H$ $\quad\mathbf{v}_t=\mathbf{R}_{t-1}\mathbf{x}_t$ ($\mathbf{Q}$ is the fourth order cumulant matrix) [11, 12]
non-stationary and decorrelated [9, 12]	$E[\mathbf{x}_k\mathbf{x}_k^H]$	$\mathbf{R}_t=\frac{t-1}{t}\mathbf{R}_{t-1}+\frac{1}{t}\mathbf{x}_t\mathbf{x}_t{}^H$ $\quad\mathbf{Q}_k=\frac{k-1}{k}\mathbf{Q}_{k-1}+\frac{1}{k}\mathbf{x}_k\mathbf{x}_k{}^H$ ($k=1$, $t=1$ are the beginning times for the non-overlapping windows on the data)
non-white and deocorrelated [10, 12]	$E[\mathbf{x}_t\mathbf{x}_{t+\tau}^H+\mathbf{x}_{t+\tau}\mathbf{x}_t^H]$	$\mathbf{R}_t=\frac{t-1}{t}\mathbf{R}_{t-1}+\frac{1}{t}\mathbf{x}_t\mathbf{x}_t{}^H$ $\quad\mathbf{Q}_t=\frac{t-1}{t}\mathbf{Q}_{t-1}+\frac{1}{t}(\mathbf{x}_t\mathbf{x}_{t+\tau}^H+\mathbf{x}_{t+\tau}\mathbf{x}_t^H)$ (τ chosen for non-zero autocorrelation in sources)
Deflation and fixed point eigendecomposition steps [13] $\mathbf{D}=\mathbf{R}^{-1}\mathbf{Q}$ $\quad\mathbf{D}_d=\left[\mathbf{I}-\frac{\mathbf{w}_{d-1}\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}}{\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}\mathbf{w}_{d-1}}\right]\mathbf{D}_{d-1}$ $\quad\mathbf{Q}_d=\left[\mathbf{I}-\frac{\mathbf{Q}_{d-1}\mathbf{w}_{d-1}\mathbf{w}_{d-1}^H}{\mathbf{w}_{d-1}^H\mathbf{Q}_{d-1}\mathbf{w}_{d-1}}\right]\mathbf{Q}_{d-1}$ $\quad\mathbf{R}_d=\mathbf{R}_{d-1}$ $\quad\mathbf{w}_d\leftarrow\frac{\mathbf{w}_d^H\mathbf{R}_d\mathbf{w}_d}{\mathbf{w}_d^H\mathbf{Q}_d\mathbf{w}_d}\mathbf{D}_d\mathbf{w}_d$ (fixed point iteration for each dimension, *d*)		

Table 1: Summary of recursive blind source separation (RBSS) algorithms.

the optimal solution based on its separation criterion in a small number of samples/iterations, even with random initialization.

7. REFERENCES

[1] A. Hyvarinen, J. Karhunen, *Independent Component Analysis*, Wiley, New York, 1991.

[2] A. Cichoki, S. I. Amari, *Adaptive Blind signal and Image Processing: Leaning Algorithms and Applications*, Wiley, New York, 2002.

[3] A. Hyvarinen, E. Oja, P. Hoyer, J. Hurri, "Image Feature Extraction by Sparse Coding and Independent Component Analysis," *Proceedings of ICPR'98*, pp. 1268-1273, 1998.

[4] T. Lan, D. Erdogmus, A. Adami, M. Pavel, "Feature Selection by Independent Component Analysis and Mutual Information Maximization in EEG Signal Classification," *Proceedings of IJCNN 2005*, pp. 3011-3016, Montreal, 2005.

[5] R. Everson, S. Roberts, "Independent Component Analysis: A Flexible Nonlineariy and Decorrelating Manifold Approach," *Neural Computation*, vol. 11, pp. 1957-1983, 2003.

[6] A. Bell, T. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Computation*, vol. 7, pp. 1129-1159, 1995.

[7] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 1996.

[8] J. Cardoso, "Blind Signal Separation: Statistical Principles," *Proceedings of the IEEE (Special Issue)*, vol. 86, pp. 2009-2025, 1998.

[9] L. Molgedey, H. G. Schuster, "Separation of a Mixture of Independent Signals Using Time Delayed Correlations," *Physical Review Letters*, vol. 72, pp. 3634-3637, 1994.

[10] E. Weinstein, M. Feder, A.V. Oppenheim, "Multi-channel signal separation by decorrelation," *IEEE Transactions on Speech Audio Processing*, vol. 1, pp. 405-413, 1993.

[11] J.-F. Cardoso, A. Souloumiac, "Blind Beamforming for Non-Gaussian Signals," *IEE Proceedings-F*, vol. 140, pp. 362-370, 1993.

[12] L. Parra, P. Sajda, "Blind Source Separation via Generalized Eigenvalue Decomposition," *Journal of Machine Learning Research*, vol.4, pp. 1261-1269, 2003.

[13] Y. N. Rao, J. C. Principe, and T. F. Wong, "Fast RLS-like algorithm for generalized eigendecomposition and its applications," *Journal of VLSI Signal Processing*, vol. 37, pp. 333-344, June 2004.

[14] A. Hyvarinen, E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis," *Neural Computation*, vol. 9, pp. 1483-1492, 1997.

[15] K.E. Hild II, D. Erdogmus, J.C. Principe, "Blind Source Separation Using Renyi's Mutual Information," *IEEE Signal Processing Letters*, vol. 8, no. 6, pp. 174-176, 2001.

[16] U. Ozertem, D. Erdogmus, T. Lan, "Recursive Generalized Eigendecomposition for Independent Components Analysis," *Proceedings of ICA 2006*, pp. 198-205, Charleston, South Carolina, USA, 2006.