NORMALISED NATURAL GRADIENT ALGORITHM FOR THE SEPARATION OF CYCLOSTATIONARY SOURCES

M. G. Jafari, and J. A. Chambers

Centre for Digital Signal Processing Research, King's College London, London WC2R 2LS, U.K.
E-mail: maria.jafari@kcl.ac.uk

ABSTRACT

A normalised natural gradient algorithm (NGA) for the separation of cyclostationary source signals is proposed in this paper. It improves the convergence properties of the cyclostationary natural gradient algorithm (CSNGA) by employing a gradient adaptive learning rate whose value changes in response to some change in the filter parameters. Experimental results demonstrate the improved behaviour of the approach.

1. INTRODUCTION

The objective of blind source separation (BSS) is to recover the original independent source signals given only a set of observations which arise when the sources are mixed by passage through some unknown medium. Although the term blind indicates that no knowledge is available about either the sources or the mixing channel, to make the problem more tractable, several assumptions are typically made regarding both. In this paper, we assume that the source signals are wide sense cyclostationary, which implies that the mean and autocorrelation function of the data vary periodically with time, and arise when the underlying process generating the signal has oscillatory behaviour, as does biomedical data, which frequently originate from breathing or contraction of the cardiac muscle, or due to modulation in man-made signals. In particular, we propose a normalised version of the CSNGA approach [1], a sequential algorithm for the separation of cyclostationary sources, based on NGA [2].

Thus, we begin with stating the BSS problem in Section 2, followed by a brief description of the CSNGA algorithm in Section 3. The normalised cyclostationary NGA algorithm is presented in Section 4, where it is also generalised to the case of complex valued sources. The performance of the proposed approach for real and complex data is shown by simulation in section 5, while conclusions are drawn in section 6.

2. PROBLEM STATEMENT

The m observed signals generated when n sources are mixed by a time-invariant instantaneous channel, and no noise is present, are given by [2]

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) \tag{1}$$

where $\mathbf{x}(k) \in \mathbb{C}^m$ is the vector of observed signals, and $\mathbf{s}(k) \in \mathbb{C}^n$ is the vector of source signals, assumed to be zero-mean and mutually independent. $\mathbf{A} \in \mathbb{C}^{m \times n}$ is an unknown, full column rank mixing matrix, and typically it is assumed that there are at least as many sensors as sources, that is $m \geq n$. The sources are recovered using the following linear separating system

$$\mathbf{y}(k) = \mathbf{W}(k)\mathbf{x}(k) \tag{2}$$

where $\mathbf{y}(k) \in \mathbb{C}^n$ is an estimate of $\mathbf{s}(k)$, and $\mathbf{W}(k) \in \mathbb{C}^{n \times m}$ is the separating matrix. The sources can only be recovered up to a multiplicative constant, and their order cannot be pre-determined, so that perfect separation is achieved when the global mixing-separating matrix, defined as

$$\mathbf{P}(k) = \mathbf{W}(k)\mathbf{A} \tag{3}$$

tends toward a matrix with only one non-zero term in each row and column [2], and is given by

$$\mathbf{P}(k) = \mathbf{J}\mathbf{D} \tag{4}$$

where $\mathbf{J} \in \mathbb{C}^{n \times n}$ is a permutation matrix modeling the ordering ambiguity, and $\mathbf{D} \in \mathbb{C}^{n \times n}$ is a diagonal matrix accounting for the scaling indeterminacy. The performance of a BSS method can be assessed by plotting the following performance index (PI)

$$\begin{aligned}\mathbf{PI}(k) = {} & \frac{1}{m}\sum_{i=1}^{m}\left\{\sum_{j=1}^{m}\frac{|p_{ij}|^2}{max_q|p_{iq}|^2 - 1}\right\} \\ & + \frac{1}{m}\sum_{j=1}^{m}\left\{\sum_{i=1}^{m}\frac{|p_{ij}|^2}{max_q|p_{qj}|^2 - 1}\right\}\end{aligned} \tag{5}$$

This work was supported by the Engineering and Physical Sciences Research Council of the U.K.

0-7803-7663-3/03/$17.00 ©2003 IEEE

where $\mathbf{P}(k) = [p_{ij}]$, and m is the number of source signals. Thus, the performance index is a measure of the closeness between $\mathbf{W}(k)$ and the pseudo-inverse of the mixing matrix, taking into account the scaling and ordering ambiguities. Generally, a low PI indicates better performance. Conventional BSS assumes that at most one source has Gaussian distribution because, for Gaussian random variables, uncorrelatedness corresponds to independence. In this paper we further assume that the sources are cyclostationary.

3. CYCLOSTATIONARY NATURAL GRADIENT ALGORITHM

The cyclostationary natural gradient algorithm attempts to minimise the following cost function [1]

$$KL\left(\mathbf{W}\left(k\right)\right) = -\log\det\left(\mathbf{W}\left(k\right)\right) - \sum_{i=1}^{m} \log q_i\left(y_i\left(k\right)\right) \\ +\frac{1}{2}Tr\left(\tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right) - \frac{1}{2}\log\det\left(\tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right) - \frac{m}{2} \qquad (6)$$

where $Tr\left(\cdot\right)$ and $\det\left(\cdot\right)$ are respectively the trace and determinant operators, and $q_i\left(y_i\left(k\right)\right)$ is an appropriately chosen independent pdf. The term $\tilde{\mathbf{R}}_y^{\alpha}\left(k\right)$ is defined as $\tilde{\mathbf{R}}_y^{\alpha}\left(k\right) = \sum_{p=1}^{m} \mathbf{R}_y^{\alpha_p}\left(k\right)$, where $\mathbf{R}_y^{\alpha_p}\left(k\right) = E\left\{e^{j\alpha_p k}\mathbf{y}\left(k\right)\mathbf{y}^T\left(k\right)\right\}$ represents the output cyclic correlation matrix for the p-th cycle frequency which is required to satisfy

$$\lim_{k\to\infty} E\left\{\mathbf{R}_y^{\alpha_p}\left(k\right)\right\} = \mathbf{I}' \qquad (7)$$

where the elements of $\mathbf{I}'$, $[\mathbf{I}']_{l,g}$ are defined by

$$[\mathbf{I}']_{l,g} = \begin{cases} 1, & \text{if } l \in \{1, 2, \ldots, m\}, g = l = p \\ 0, & \text{otherwise} \end{cases} \qquad (8)$$

Then, in the limit as $k \to \infty$, each of the output cyclic correlation matrices converges to a matrix with only one non-zero entry, situated at the p-th position along the main diagonal, giving $\lim_{k\to\infty} \tilde{\mathbf{R}}_y^{\alpha}\left(k\right) = \mathbf{I}$. When the source signals and the mixing matrix are real valued, the update equation for the cyclostationary natural gradient algorithm is given by

$$\mathbf{W}\left(k+1\right) = \mathbf{W}\left(k\right) + \mu\left(k\right)\left[\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) \right. \\ \left. +\mathbf{I} - \tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right]\mathbf{W}\left(k\right) \qquad (9)$$

where $\mu\left(k\right)$ is the learning rate. Reasoning along the lines of [3], the learning rule (9) effectively represents a single stage sequential algorithm performing second- and higher-order conventional decorrelation simultaneously, as well as second-order cyclic decorrelation.

4. ADAPTIVE STEP-SIZE PARAMETER

The use of a fixed step-size parameter generally leads to slow convergence speed and poor tracking performance. Alternatively, a time-varying step-size parameter can be employed, which changes in response to some change in the filter parameters. Hence, based on the method outlined in [4], a gradient adaptive step-size algorithm is derived, which updates the learning rate so that at every iteration it attempts to minimise the CSNGA cost function (6). Thus, the learning rate at time k is evaluated recursively according to [4]

$$\mu\left(k\right) = \mu\left(k-1\right) - \rho\frac{\partial KL\left(\mathbf{W}\left(k\right)\right)}{\partial\mu\left(k-1\right)} \qquad (10)$$

where ρ is a step-size parameter. As in [4], we assume for the sake of clarity that there are as many source as there are mixtures $m = n$, and that, for small learning rates

$$\mathbf{W}\left(k\right)\mathbf{x}\left(k+1\right) \approx \mathbf{y}\left(k+1\right) \qquad (11)$$

Differentiating (6) with respect to $\mu\left(k\right)$ gives

$$\frac{\partial KL\left(\mathbf{W}\left(k+1\right)\right)}{\partial\mu\left(k\right)} = -\frac{\partial\log\det\left(\mathbf{W}\left(k+1\right)\right)}{\partial\mu\left(k\right)} \\ -\sum_{i=1}^{m}\frac{\partial\log q_i\left(y_i\left(k+1\right)\right)}{\partial y_i\left(k+1\right)}\frac{\partial y_i\left(k+1\right)}{\partial\mu\left(k\right)} \\ +\frac{1}{2}\frac{\partial Tr\left(\tilde{\mathbf{R}}_y^{\alpha}\left(k+1\right)\right)}{\partial\mu\left(k\right)} - \frac{1}{2}\frac{\partial\log\det\left(\tilde{\mathbf{R}}_y^{\alpha}\left(k+1\right)\right)}{\partial\mu\left(k\right)} \qquad (12)$$

Substituting (9) into the first term on the right-hand side of (12) we have

$$\frac{\partial\log\det\mathbf{W}\left(k+1\right)}{\partial\mu\left(k\right)} = \frac{\partial\log\det}{\partial\mu\left(k\right)}\left\{\mathbf{I} + \mu\left(k\right)\left[\mathbf{I}\right.\right. \\ \left.\left. -\mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right]\right\} + \frac{\partial\log\det\mathbf{W}\left(k\right)}{\partial\mu\left(k\right)} \qquad (13)$$

To compute the differential in (13), the determinant of the matrix must be evaluated $\left\{\mathbf{I} + \mu\left(k\right)\left[\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right]\right\}$. This can be achieved using the result that the determinant of an $m \times m$ matrix equals the product of its m eigenvalues [5]. Then $\left\{\mathbf{I} + \mu\left(k\right)\left[\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^{\alpha}\left(k\right)\right]\right\}$ has $m-1$ eigenvalues equal to $1 + 2\mu\left(k\right)$, and one equal to $1+\mu\left(k\right)\left[2 - \mathbf{y}^T\left(k\right)\mathbf{f}\left(\mathbf{y}\left(k\right)\right) - \sum_{p=1}^{m} e^{j\alpha_p k} \times \mathbf{y}^T\left(k\right)\mathbf{y}\left(k\right)\right]$ [4], because the matrices $\mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k)$ and $\tilde{\mathbf{R}}_y^{\alpha}\left(k\right)$ are rank deficient, both of rank 1. Thus, letting $e^{j\alpha k} = \sum_{p=1}^{m} e^{j\alpha_p k}$, (13) becomes

$$\frac{\partial\log\det\mathbf{W}\left(k+1\right)}{\partial\mu\left(k\right)} = \frac{2\left(m-1\right)}{\left(1+2\mu\left(k\right)\right)} \\ +\frac{\left(2 - \mathbf{y}^T\left(k\right)\mathbf{f}\left(\mathbf{y}\left(k\right)\right) - e^{j\alpha k}\mathbf{y}^T\left(k\right)\mathbf{y}\left(k\right)\right)}{1 + \mu\left(k\right)\left(2 - \mathbf{y}^T\left(k\right)\mathbf{f}\left(\mathbf{y}\left(k\right)\right) - e^{j\alpha k}\mathbf{y}^T\left(k\right)\mathbf{y}\left(k\right)\right)} \qquad (14)$$

which holds for $0 < \mu(k) \ll |2- \mathbf{y}^T(k)\,\mathbf{f}(\mathbf{y}(k)) -e^{j\alpha k}\mathbf{y}^T(k)\,\mathbf{y}(k)|$ [4].

A similar approach is followed to evaluate the fourth term of (12), in which, from (2), $\tilde{\mathbf{R}}_y^\alpha(k+1)$ can be replaced with $\mathbf{W}(k+1)\,\tilde{\mathbf{R}}_x^\alpha(k+1)\,\mathbf{W}^T(k+1)$. Also, assuming that, for small learning rates

$$\mathbf{W}(k)\,\tilde{\mathbf{R}}_x^\alpha(k+1)\,\mathbf{W}^T(k+1) \approx \tilde{\mathbf{R}}_y^\alpha(k+1) \qquad (15)$$

the derivative of $\log\det\left(\tilde{\mathbf{R}}_y^\alpha(k+1)\right)$ is given by

$$\frac{\partial \log\det\left(\tilde{\mathbf{R}}_y^\alpha(k+1)\right)}{\partial\mu(k)} = \frac{2(m-1)}{(1+2\mu(k))} + \frac{\left(2-\mathbf{y}^T(k)\,\mathbf{f}(\mathbf{y}(k)) - e^{j\alpha k}\mathbf{y}^T(k)\,\mathbf{y}(k)\right)}{1+\mu(k)\left(2-\mathbf{y}^T(k)\,\mathbf{f}(\mathbf{y}(k)) - e^{j\alpha k}\mathbf{y}^T(k)\,\mathbf{y}(k)\right)} \qquad (16)$$

To evaluate the second term on the right-hand side of (12), we use $f_i(y_i) = -\frac{\partial \log q_i(y_i)}{\partial y_i}$ [2] giving

$$-\sum_{i=1}^{m} \frac{\partial \log q_i(y_i(k+1))}{\partial y_i(k+1)}\frac{\partial y_i(k+1)}{\partial\mu(k)} = \sum_{i=1}^{m} f_i(y_i(k+1)) \times \frac{\partial y_i(k+1)}{\partial\mu(k)} \qquad (17)$$

From (2) and (9), and using the approximation (11) we have

$$\mathbf{y}(k+1) \approx \mathbf{W}(k)\,\mathbf{x}(k+1) + \mu\,[\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k)\Big]\,\mathbf{y}(k+1) \qquad (18)$$

Differentiating and pre-multiplying by $\mathbf{f}^T(\mathbf{y}(k+1))$, gives

$$\mathbf{f}^T(\mathbf{y}(k+1))\,\frac{\partial\mathbf{y}(k+1)}{\partial\mu(k)} \approx \mathbf{f}^T(\mathbf{y}(k+1))\,[\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k)\Big]\,\mathbf{y}(k+1) \qquad (19)$$

Finally, differentiating $Tr\left(\tilde{\mathbf{R}}_y^\alpha(k+1)\right)$ with respect to $\mu(k)$ yields

$$\frac{\partial Tr\left(\tilde{\mathbf{R}}_y^\alpha(k+1)\right)}{\partial\mu(k)} \approx Tr\,([\mathbf{I} - \mathbf{f}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k)\Big]\,\tilde{\mathbf{R}}_y^\alpha(k+1)\Big) \qquad (20)$$

where (9) and (15) have been used. Substituting back into (10), and letting $b(k-1) = (2-\mathbf{y}^T(k-1)\,\mathbf{f}(\mathbf{y}(k-1)) -e^{j\alpha k}\mathbf{y}^T(k-1)\,\mathbf{y}(k-1))$, the new gradient adaptive step-size algorithm is given by

$$\begin{aligned}\mu(k) = \mu(k-1) + \rho\Big\{&\mathbf{f}^T(\mathbf{y}(k))\,\tilde{\mathbf{R}}_y^\alpha(k-1)\,\mathbf{y}(k) \\ &-2\mathbf{y}^T(k)\,\mathbf{f}(\mathbf{y}(k)) + \mathbf{f}^T(\mathbf{y}(k))\,\mathbf{f}(\mathbf{y}(k-1)) \\ &\times\mathbf{y}^T(k-1)\mathbf{y}(k) + \frac{3(m-1)}{(1+2\mu(k-1))} \\ &+\frac{3b(k-1)}{2[1+\mu(k)\,b(k-1)]} - \frac{1}{2}Tr\,([\mathbf{I} - \mathbf{f}(\mathbf{y}(k-1)) \\ &\times\mathbf{y}^T(k-1) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k-1)\Big]\,\tilde{\mathbf{R}}_y^\alpha(k)\Big)\Big\}\end{aligned} \qquad (21)$$

As in [4], $\mu(k) \in [\delta, \mu_{\max}]$, where δ is a small positive constant preventing the adaptation of the learning rate from terminating entirely, and $\mu_{\max}$ represents an upper bound, controlling the size of $\mu(k)$, thus ensuring stability of (21). Moreover, it is reported in [4] that, as with all gradient step-size methods, the greatest disadvantage of the above algorithm is the need to select an appropriate step-size parameter ρ, although simulations have shown in general that gradient step-size algorithms are relatively insensitive to its value. This applies also to (21).

4.1. Complex CSNGA (CCSNGA)

Extension of the cyclostationary NGA algorithm to the complex case is readily achieved by modifying (9), such that the transpose operator is replaced by the Hermitian transpose operator, and an appropriate phase preserving complex activation function $\mathbf{g}(\mathbf{y}(k))$ is selected [6]. A common strategy is to employ so called split-complex non-linearities, such as $g_i(y_i(k)) = \tanh(y_{iR}(k)) + j\tanh(y_{iI}(k))$ where $y_{iR}(k)$ and $y_{iI}(k)$ denote respectively the real and imaginary parts of $y_i(k)$, when the sources are super-Gaussian, or as [3] $g_i(y_i(k)) = y_i(k)\,|y_i|^2$ for the sub-Gaussian case. Thus, the learning rule (9) becomes

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu(k)\,\Big[\mathbf{I} - \mathbf{g}(\mathbf{y}(k))\mathbf{y}^H(k) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k)\Big]\,\mathbf{W}(k) \qquad (22)$$

Extension of the adaptive step-size algorithm to the complex case follows from the same rules as above

$$\begin{aligned}\mu(k) = \mu(k-1) + \rho\mathbb{R}e\Big\{&\mathbf{g}^H(\mathbf{y}(k))\,\tilde{\mathbf{R}}_y^\alpha(k-1)\,\mathbf{y}(k) \\ &-2\mathbf{y}^H(k)\,\mathbf{g}(\mathbf{y}(k)) + \mathbf{g}^H(\mathbf{y}(k))\,\mathbf{g}(\mathbf{y}(k-1)) \\ &\times\mathbf{y}^H(k-1)\mathbf{y}(k) + \frac{3(m-1)}{(1+2\mu(k-1))} \\ &+\frac{3b(k-1)}{2[1+\mu(k)\,b(k-1)]} - \frac{1}{2}Tr\,([\mathbf{I} - \mathbf{g}(\mathbf{y}(k-1)) \\ &\times\mathbf{y}^H(k-1) + \mathbf{I} - \tilde{\mathbf{R}}_y^\alpha(k-1)\Big]\,\tilde{\mathbf{R}}_y^\alpha(k)\Big)\Big\}\end{aligned} \qquad (23)$$

where $\mathbb{R}e\{u\}$ represents the real part of u. The learning rate must remain real valued to ensure the descent direction is not modified.

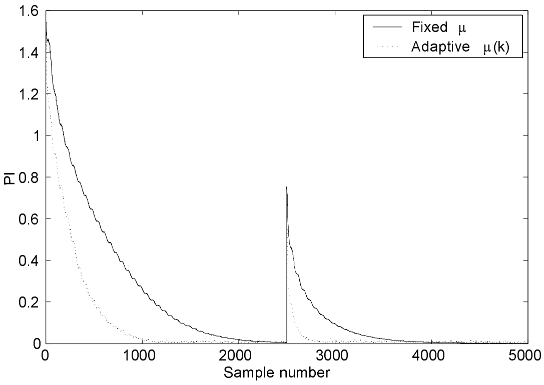

Fig. 1. Average PIs obtained with CSNGA, for real sources and mixing matrix, and with fixed and adaptive learning rate.

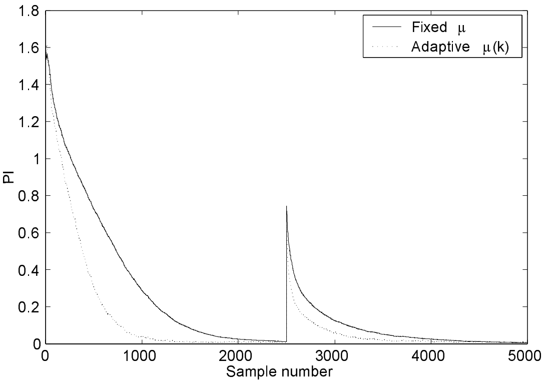

Fig. 2. Average PIs obtained with CSNGA, for complex sources and a real mixing matrix, together with fixed and adaptive learning rate.

5. SIMULATIONS

The behaviour of the CSNGA algorithm with gradient adaptive step-size is illustrated by computer simulation. The sources are a sinusoidal noise signal of normalised frequency $(40\pi)^{-1}$, and two BPSK signals carrying independent binary data, and using sinusoidal carriers of frequencies $2(5\pi)^{-1}$ and $(4\pi)^{-1}$. The sources are mixed by a real matrix $\mathbf{A}$ for $0 \leq k < 2500$, and by its transpose $\mathbf{A}^T$ for $k \geq 2500$, and zero mean white Gaussian noise is added, so that the SNR is 10 dB. Since the source signals and mixing matrix are real, the exponential function in equation (9) simplifies to a cosine function. The resulting mixtures are separated using CSNGA with fixed step-size parameter $\hat{\mu}$ equal to 0.001, and with the adaptive learning rate in (21), where $\mu(0) = 0.001$, $\rho = 10^{-7}$, $\delta = 10^{-4}$ and $\mu_{\max} = 0.005$. Figure 1 shows that the average performance of the CSNGA algorithm improves considerably when the adaptive step-size method is employed, since the algorithm reacts quickly to the changes in the mixing channel.

Next, two QPSK signals modulated by sinusoids of carrier frequencies $2(5\pi)^{-1}$ and $(4\pi)^{-1}$, and one complex sinusoid of frequency $(40\pi)^{-1}$ are mixed by a real instantaneous mixing channel which changes abruptly after 2500 samples. Complex valued circular zero mean white Gaussian noise is added such that the SNR is 10 dB. Separation is carried out using the complex CSNGA method, when the learning rate is fixed $\hat{\mu} = 0.002$, and when the adaptive step-size (23) is used, with $\mu(0) = 0.002$, $\rho = 10^{-7}$, $\delta = 10^{-4}$ and $\mu_{\max} = 0.005$. Figure 2 shows that the adaptive step-size method tracks the changes in the mixing channel more quickly than the fixed step-size approach. In particular, the initial convergence speed, as well as the speed of convergence following the abrupt change, is found to increase.

6. CONCLUSIONS

A normalised natural gradient algorithm for the separation of real and complex valued cyclostationary sources has been presented. Simulation results have shown that the algorithm leads to fast speed of convergence for both real and complex valued sources, and when the mixing matrix changes abruptly, and in general it improves the convergence properties of the CSNGA approach.

7. REFERENCES

[1] M. G. Jafari, J. A. Chambers, and D. P. Mandic, “Natural gradient algorithm for cyclostationary sources,” *IEE Electronics Letters*, vol. 38, pp. 758–759, 2002.

[2] S. Amari and A. Cichocki, “Adaptive blind signal processing - neural network approaches,” *Proceedings of the IEEE*, vol. 86, pp. 2026–2048, 1998.

[3] J. F. Cardoso and B. Laheld, “Equivariant adaptive source separation,” *IEEE Trans. on Signal Processing*, vol. 44, pp. 3017–3030, 1996.

[4] S. C. Douglas and A. Cichocki, “Adaptive step size techniques for decorrelation and blind source separation,” *Proc. of the Asilomar Conf. on Signals, Systems and Computers*, vol. 2, pp. 1191–1195, 1998.

[5] G. Strang, *Linear algebra and its applications*. Harcourt Brace Jovanovich, 3rd ed., 1988.

[6] H. M. Park, H. Y. Jung, T. W. Lee, and S. Y. Lee, “On subband-based blind signal separation for noisy speech recognition,” *Electronic Letters*, vol. 35, pp. 2011–2012, 1999.