

# Linear Prediction Based Blind Source Extraction Algorithms in Practical Applications

Zhi-Lin Zhang<sup>1,2</sup> and Liqing Zhang<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Shanghai Jiao Tong University, Shanghai 200240, China

<sup>2</sup> School of Computer Science and Engineering,  
University of Electronic Science and Technology of China,  
Chengdu 610054, China  
zllzhang@uestc.edu.cn, zhang-lq@cs.sjtu.edu.cn

**Abstract.** Blind source extraction (BSE) is of advantages over blind source separation (BSS) when obtaining some underlying source signals from high dimensional observed signals. Among a variety of BSE algorithms, a large number of algorithms are based on linear prediction (LP-BSE). In this paper we analyze them from practical point of view. We reveal that they are, in nature, minor component analysis (MCA) algorithms, and thus they have some problems that are inherent in MCA algorithms. We also find a switch phenomenon of online LP-BSE algorithms, showing that different parts of a single extracted signal are the counterparts of different source signals. The two issues should be noticed when one applies these algorithms to practical applications. Computer simulations are given to confirm these observations.

## 1 Introduction

Blind source extraction (BSE) [1] is a powerful technique that is closely related to blind source separation (BSS). The basic task of BSE is to estimate some of underlying source signals that are linearly combined in observations. Compared with BSS, BSE has some advantages [1]. An attractive one is its ability to extract a small subset of source signals from high-dimensional observed signals. Hence it is often recommended to be used in EEG/MEG fields and alike [1–3].

There are many BSE algorithms for extracting source signals based on their temporal structures [1, 4]. Among them there is a class of algorithms based on linear prediction. For example, Cichocki, Mandic, and Liu et al. [2, 6–8] proposed several BSE algorithms based on short-term linear prediction. Barros et al. [5] proposed a BSE algorithm based on long-term linear prediction. Later Smith et al. [3] proposed a BSE algorithm combining short-term prediction and long-term prediction. And recently Liu et al.[9] extended a basic linear prediction based algorithm to the one suitable for noisy environment.

In this paper we consider some possible problems when applying the linear prediction based BSE (LP-BSE) algorithms to practical applications, especially EEG/MEG fields.

## 2 The Linear Prediction Based Algorithms

Suppose that unknown source signals  $\mathbf{s}(k) = [s_1(k), \dots, s_n(k)]^T$  are zero-mean and spatially uncorrelated, and suppose that  $\mathbf{x}(k) = [x_1(k), \dots, x_n(k)]^T$  is a vector of observed signals, which is a linear instantaneous mixture of source signals by  $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$ , where  $k$  is time index and  $\mathbf{A} \in \mathbf{R}^{n \times n}$  is an unknown mixing matrix of full rank. The goal of BSE is to find a demixing vector  $\mathbf{w}$  such that  $y(k) = \mathbf{w}^T \mathbf{x}(k) = \mathbf{w}^T \mathbf{A}\mathbf{s}(k)$  is an estimate of a source signal. To cope with ill-conditioned cases and to make algorithms simpler and faster, before extraction whitening [1] is often used to transform the observed signals  $\mathbf{x}(k)$  to  $\mathbf{z}(k) = \mathbf{V}\mathbf{x}(k)$  such that  $E\{\mathbf{z}(k)\mathbf{z}(k)^T\} = \mathbf{I}$ , where  $\mathbf{V} \in \mathbf{R}^{n \times n}$  is a prewhitening matrix and  $\mathbf{V}\mathbf{A}$  is an orthogonal matrix.

Assuming that the underlying source signals have temporal structures, the class of LP-BSE algorithms is derived by minimizing the normalized mean square prediction error given by [6, 8]

$$J_1 = \frac{E\{e(k)^2\}}{E\{y(k)^2\}} = \frac{E\{(y(k) - \mathbf{b}^T \mathbf{y}(k))^2\}}{E\{y(k)^2\}} \quad (1)$$

where  $y(k) = \mathbf{w}^T \mathbf{x}(k)$ ,  $\mathbf{b} = [b_1, b_2, \dots, b_P]^T$ ,  $\mathbf{y}(k) = [y(k-1), y(k-2), \dots, y(k-P)]^T$  and  $P$  is AR order that is set before running algorithms. If one performs the whitening and normalizes the demixing vector  $\mathbf{w}$ , the objective function (1) reduces to [2, 5]:

$$J_2 = E\{e(k)^2\} = E\{(y(k) - \mathbf{b}^T \mathbf{y}(k))^2\} \quad (2)$$

where  $y(k) = \mathbf{w}^T \mathbf{z}(k) = \mathbf{w}^T \mathbf{V}\mathbf{x}(k)$  and  $\|\mathbf{w}\| = 1$ .

Without loss of generality, we only consider the objective function (2) in the following. After some algebraic calculations, from (2) we obtain

$$J_2 = E\{e(k)^2\} = \mathbf{w}^T \widehat{\mathbf{R}}_{\mathbf{Z}} \mathbf{w} = \mathbf{w}^T \mathbf{V} \mathbf{A} \widehat{\mathbf{R}}_{\mathbf{S}} \mathbf{A}^T \mathbf{V}^T \mathbf{w} = \mathbf{q}^T \widehat{\mathbf{R}}_{\mathbf{S}} \mathbf{q}, \quad (3)$$

in which  $\mathbf{q} = \mathbf{A}^T \mathbf{V}^T \mathbf{w}$ , and

$$\widehat{\mathbf{R}}_{\mathbf{Z}} = \mathbf{R}_{\mathbf{Z}}(0) - \sum_{p=1}^P b_p \mathbf{R}_{\mathbf{Z}}(p) - \sum_{q=1}^P b_q \mathbf{R}_{\mathbf{Z}}(-q) + \sum_{p=1}^P \sum_{q=1}^P b_p b_q \mathbf{R}_{\mathbf{Z}}(q-p) \quad (4)$$

$$\widehat{\mathbf{R}}_{\mathbf{S}} = \mathbf{R}_{\mathbf{S}}(0) - 2 \sum_{p=1}^P b_p \mathbf{R}_{\mathbf{S}}(p) + \sum_{p=1}^P \sum_{q=1}^P b_p b_q \mathbf{R}_{\mathbf{S}}(q-p) \quad (5)$$

where  $\mathbf{R}_{\mathbf{Z}}(p) = E\{\mathbf{z}(k)\mathbf{z}(k-p)^T\}$ , and  $\mathbf{R}_{\mathbf{S}}(p) = E\{\mathbf{s}(k)\mathbf{s}(k-p)^T\}$  is a diagonal matrix due to the assumptions. Also,  $\widehat{\mathbf{R}}_{\mathbf{S}}$  is a diagonal matrix, whose diagonal elements are given by

$$\rho_i = r_i(0) - 2 \sum_{p=1}^P b_p r_i(p) + \sum_{p=1}^P \sum_{q=1}^P b_p b_q r_i(q-p), \quad i = 1, \dots, n \quad (6)$$

where  $r_i$  is the autocorrelation function of  $s_i$ .

Now we calculate the concrete value of  $\rho_i$ . Suppose when  $J_2$  achieves its minimum,  $\mathbf{b}$  achieves  $\mathbf{b}^* = [b_1^*, b_2^*, \dots, b_p^*]^T$ . We express all the source signals as

$$s_i(k) = \sum_{p=1}^P b_p^* s_i(k-p) + e_i(k), \quad i = 1, \dots, n \quad (7)$$

where  $e_i(k)$  is called residual processes. Then we have

$$\begin{aligned} r_i(0) &= E \left\{ \left( \sum_{p=1}^P b_p^* s_i(k-p) + e_i(k) \right) \left( \sum_{q=1}^P b_q^* s_i(k-q) + e_i(k) \right) \right\} \\ &= \sum_{p=1}^P \sum_{q=1}^P b_p^* b_q^* r_i(q-p) + 2E\{e_i(k)s_i(k)\} - E\{e_i(k)^2\} \end{aligned} \quad (8)$$

where we use the relationship (7). On the other hand, we also have

$$r_i(0) = E \left\{ \left( \sum_{p=1}^P b_p^* s_i(k-p) + e_i(k) \right) s_i(k) \right\} = \sum_{p=1}^P b_p^* r_i(p) + E\{e_i(k)s_i(k)\}. \quad (9)$$

Substitute (8) and (9) into (6), we obtain

$$\rho_i = E\{e_i(k)^2\}, \quad (10)$$

implying that  $\rho_i (i = 1, \dots, n)$  are just the powers of residual processes of linear prediction to the source signals given the coefficients  $b_p^* (p = 1, \dots, P)$ . Obviously, calculating the minimum of  $J_2$  is equivalently finding the minimum among all  $\rho_i (i = 1, \dots, n)$ , which are the eigenvalues of  $\hat{\mathbf{R}}_{\mathbf{S}}$  and are also the ones of  $\hat{\mathbf{R}}_{\mathbf{Z}}$ . And the demixing vector  $\mathbf{w}$  is the associated eigenvector. Thus the LP-BSE algorithms are in nature the MCA algorithms [10, 11, 15].

### 3 Analysis of the LP-BSE Algorithms

It is recognized that MCA algorithms have some flaws in practical applications [10, 11]. First, in practice the small eigenvalues of the covariance matrix  $\hat{\mathbf{R}}_{\mathbf{Z}}$  are often close to each other, which reduces the estimate accuracy of associated eigenvectors [14] and brings difficulties to global convergence [10, 11]. Moreover the performance of MCA algorithms often suffers from outliers and noise [12].

Naturally, the LP-BSE algorithms inherit some of these flaws when dealing with high dimensional observed signals. Take the extraction of event-related potentials as an example. The number of sensor signals are often larger than 64, and some underlying source signals have similar time structures [13]. According to (10) and (7) the small eigenvalues of  $\hat{\mathbf{R}}_{\mathbf{Z}}$  are close to each other, which makes the estimation of the minor eigenvector sensitive to sensor noise [12].

Now consider online versions of LP-BSE algorithms. Suppose the current extracted source is  $s_1(k)$ , whose current residual process's power level is  $e_1^2(k)$ . This implies that given the prediction AR order  $P$  in algorithms,  $e_1^2(k)$  is the smallest among all  $e_j^2(k), j = 1, \dots, n$ . If at time  $k+1$ ,  $s_1(k+1)$ 's true AR order starts to change but the given prediction AR order does not change,  $e_1^2(k+1)$  may become larger<sup>3</sup>. Then there may be another source signal, say  $s_2(k+1)$ , whose  $e_2^2(k+1)$  with the given prediction order is smaller than that of  $s_1(k+1)$ . Consequently, the algorithms switch to extract  $s_2(k+1)$ . Therefore the extracted signal is still mixed by the two source signals in the sense that the first part of the extracted signal is the counterpart of  $s_1$  and the second part is the counterpart of  $s_2$ . We call this the switch phenomenon. The essential reason to the existence of the switch phenomenon is the use of the fixed prediction order that is set before performing LP-BSE algorithms. Similarly, if the true AR coefficients of source signals vary fast and  $b_i (i = 1, \dots, P)$  cannot be adjusted in the same pace, the switch phenomenon may also occur. Remind that in the EEG data processing, especially in the even-related brain potential extraction, the underlying source signals' AR order and coefficients may quickly vary. Thus the phenomenon may occur in these cases.

## 4 Simulations

In the first simulation we illustrated unsatisfying performance of LP-BSE algorithms due to their MCA nature. We used the data set ABio7, a benchmark in ICALAB [17]. Three typical LP-BSE algorithms, i.e. the ones in [2, 7, 8], were used to extract these signals. To make comparison, we intuitively gave a PCA-like BSE algorithm, a variation of our algorithm [4], as follows<sup>4</sup>:

$$\mathbf{w} = PCA_i\left(\sum_{i=1}^P \mathbf{R}_{\mathbf{z}}(\tau_i)\right) = PCA_i(\widetilde{\mathbf{R}}_{\mathbf{z}}), \quad (11)$$

where  $\mathbf{R}_{\mathbf{z}}(\tau_i) = E\{\mathbf{z}(k)\mathbf{z}(k - \tau_i)^T\}$ ,  $\tau_i$  was time delay, and  $PCA_i(\widetilde{\mathbf{R}}_{\mathbf{z}})$  was the operator that calculated the  $i$ -th principal eigenvector of  $\widetilde{\mathbf{R}}_{\mathbf{z}}$ . Using *a priori* knowledge one can choose a specific set of time delays to achieve better performance [4]. Actually, (11) is only a framework, and can be implemented offline or online by using many efficient and robust methods [14, 16]. Note that the PCA-like BSE algorithm obtains principal eigenvectors, while the LP-BSE algorithms obtain minor ones. All the algorithms were implemented offline.

The source signals were randomly mixed and whitened. Then each algorithm was performed on these signals. The step-size of the algorithm in [8] was 0.1.

<sup>3</sup> It also may become smaller. So in this case the switch phenomenon does not occur.

<sup>4</sup> Note that we present the PCA-like algorithm in purpose to show that the class of LP-BSE algorithms may not achieve satisfying results when applied to practical applications. Admittedly, better algorithms than the algorithm may be developed, which is not the topic in this paper.

The learning rate parameter  $\mu_0$  of the algorithm in [7] (see Equ.(16) in [7]) was 0.5. The extraction performance was measured by

$$PI = \frac{1}{n-1} \left( \sum_{i=1}^n \frac{q_i^2}{\max_i q_i^2} - 1 \right) \quad (12)$$

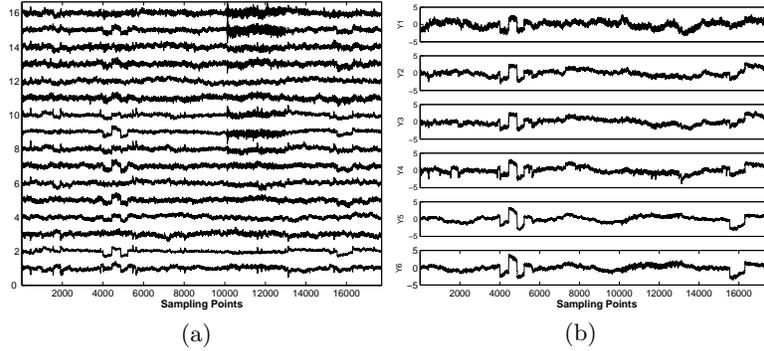
where  $\mathbf{q} = [q_1, \dots, q_n] = \mathbf{w}^T \mathbf{V} \mathbf{A}$  was a global vector,  $\mathbf{V}$  was the whitening matrix,  $\mathbf{A}$  was the mixing matrix and  $\mathbf{w}$  was the demixing vector obtained by algorithms.  $PI$ 's value lay in  $[0,1]$  for any vector  $\mathbf{q} = [q_1, \dots, q_n]$ . The smaller it was, the better the extraction performance was. Simulations were independently carried out 50 trials. The results are shown in Table 1, from which we can see that the LP-BSE algorithms generally performed poorly.

**Table 1.** The averaged performance indexes of the algorithms in the first simulation. For the three LP-BSE algorithms the parameter  $P$  was the prediction order, while for the PCA-like algorithm  $P$  meant that the time delay set was  $\{1, \dots, P\}$ .

$P$	1	2	3	4	5	6	7	8	9	10	12	20	30	40	50	400
Alg. (11)	0.00	0.00	0.01	0.09	0.02	0.07	0.01	0.01	0.06	0.02	0.01	0.05	0.04	0.07	0.03	0.01
Alg. [8]	0.19	0.17	0.17	0.18	0.18	0.18	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Alg. [2]	0.02	0.00	0.07	0.07	0.06	0.03	0.03	0.02	0.04	0.08	0.10	0.05	0.06	0.07	0.08	0.14
Alg. [7]	0.11	0.14	0.11	0.09	0.12	0.09	0.10	0.11	0.13	0.12	0.13	0.06	0.13	0.08	0.16	0.08

In the second simulation we used the 122-dimension MEG data set (Fig.1 (a)) in [18] to show performance of a typical LP-BSE algorithm in extracting horizontal eye movements, which occurred at about the 4000-th sampling point and the 15500-th sampling point. Since the movements resulted from the same group of muscles, we safely believed that artifacts associated with the movements occurring at different time should appear in the same extracted signal.

After performing the same preprocessing as that in [18], we used the offline LP-BSE algorithm in [8] to extract the artifacts with different levels of data dimension reduction. Its step-size was 0.5 and prediction order was 10. The results are shown in Fig.1 (b), where  $y_1, y_2, y_3$  and  $y_4$  were extracted by the LP-BSE algorithm with data dimension reduced to 120, 80, 60, and 40, respectively.  $y_5$  was extracted by the PCA-like algorithm (11) without data dimension reduction ( $\tau_i = \{1, \dots, 5\}$ ).  $y_6$  was extracted by FastICA, which was also used in [18]. Since Vigário et al. have shown that FastICA can perfectly extract the horizontal eye movement artifacts, we regarded  $y_6$  as a benchmark. From  $y_1 - y_3$  we see that the artifacts were not perfectly extracted, since the horizontal eye movement artifact at about the 15500-th sampling point was not extracted. Although in  $y_4$  all the artifacts were extracted, it was mixed by artifacts resulting from eye blinks [18]. Besides, we see that the extraction performance of the LP-BSE algorithm was affected by the dimension reduction. When the dimension was reduced to a certain degree, the extraction performance became relatively better. In contrast, in  $y_5$  all the horizontal eye movement artifacts were extracted without mixed



**Fig. 1.** A subset of the MEG data set [18] (a) and extracted artifacts (b).

by other artifacts, and we found the extraction quality was not affected by the data dimension (the extraction results with dimension reduction are not shown here due to limited space). We also ran other LP-BSE algorithms and almost obtained the same results. Due to space limit we omit the report.

In the last simulation we showed the switch phenomenon of online LP-BSE algorithms. We generated three AR(6) Gaussian signals of 5-second duration time (Fig.2). Each source signal had zero mean and unit variance. The sampling frequency was 1000 Hz. The AR coefficients of each signal were unchanged during the first 2.5 second, given by:

$$source1 : \mathbf{b} = [-1.6000, 0.9000, -0.2000, 0.0089, 0.0022, -0.0002]$$

$$source2 : \mathbf{b} = [-0.1000, -0.4300, 0.0970, 0.0378, -0.0130, 0.0009]$$

$$source3 : \mathbf{b} = [-2.3000, 2.0400, -0.8860, 0.1985, -0.0216, 0.0009]$$

And hereafter the AR coefficients changed to:

$$source1 : \bar{\mathbf{b}} = [-1.6000, 0.9000, -0.2000, 0.0089, 0.0022, -0.0002]$$

$$source2 : \bar{\mathbf{b}} = [-2.3000, 2.0400, -0.8860, 0.1985, -0.0216, 0.0009]$$

$$source3 : \bar{\mathbf{b}} = [-0.1000, -0.4300, 0.0970, 0.0378, -0.0130, 0.0009]$$

We used the online version of the LP-BSE algorithm in [8] to extract a signal. Its step-size was 0.01 and prediction order was 10. The result is shown in Fig.2 (see  $y_1$ ), from which we see that the first part of  $y_1$  (before 2.5 second) was the counterpart of source signal  $s_3$ , but from 3.6 second or so the signal was clearly the counterpart of source signal  $s_1$ . To further confirm this, we measured the similarity between the extracted signal and the source signals, using the performance index  $PI_2 = -10 \lg(E\{(s(k) - \tilde{s}(k))^2\})(dB)$ , where  $s(k)$  was the desired source signal, and  $\tilde{s}(k)$  was the extracted signal (both of them were normalized to be zero-mean and unit-variance). The higher  $PI_2$  is, the better the performance. Denote by *Part1* the extracted signal's segment from 2.0 s to 2.5 s, and denote by *Part2* the extracted signal's segment from 4.0 s to 5.0 s.

The  $PI_2$  of *Part1* measuring the similarity between *Part1* and the counterpart of  $s_3$  was 18.5 dB, showing *Part1* was very similar to the counterpart of  $s_3$ . The  $PI_2$  of *Part2* measuring the similarity between *Part2* and the counterpart of  $s_1$  was 19.7 dB, showing *Part2* was very similar to the counterpart of  $s_1$ .

Next we used an online version of the PCA-like algorithm (11), implemented by the OJAN PCA algorithm [16], to extract a source signal. The extracted signal is shown in Fig.2 (see  $y_2$ ), from which we can see that the extracted signal was just  $s_3$  and the switch phenomenon did not occur. We also calculated the algorithm's  $PI_2$  at *Part1* and *Part2*. The  $PI_2$  of *Part1* measuring the similarity between *Part1* and the counterpart of  $s_3$  was 22.3 dB, showing *Part1* was very similar to the counterpart of  $s_3$ . The  $PI_2$  of *Part2* measuring the similarity between *Part2* and the counterpart of  $s_3$  was 19.9 dB, showing *Part2* was very similar to the counterpart of  $s_3$  as well. The results show that the online version has well extracted the whole source signal  $s_3$ .

## 5 Conclusion

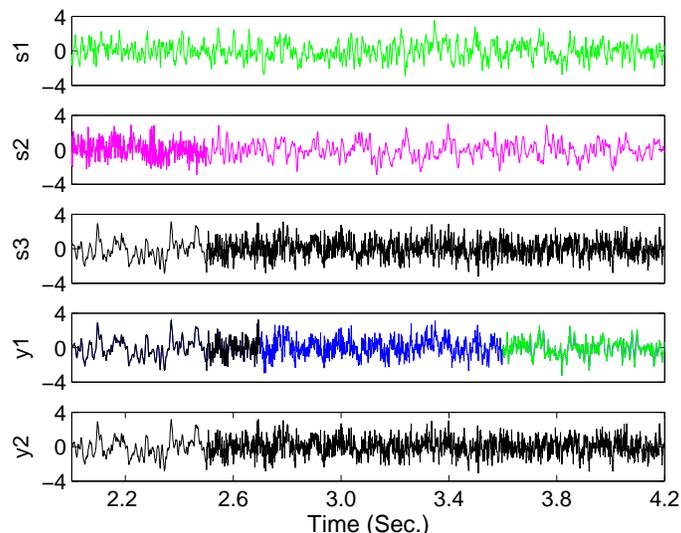
In this paper we analyze a class of linear prediction based BSE algorithms, revealing that they are in nature the MCA algorithms and showing a switch phenomenon of their online versions. Based on these results, careful attentions should be paid when one applies these algorithms to practical applications such as EEG and MEG fields.

## Acknowledgments

The work was supported by the National Basic Research Program of China (Grant No. 2005CB724301) and National Natural Science Foundation of China (Grant No.60375015).

## References

1. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, New York (2002)
2. Cichocki, A., et al.: A Blind Extraction of Temporally Correlated but Statistically Dependent Acoustic Signals. Proc. of the 2000 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing X (2000) 455-464
3. Smith, D., Lukasiak, J., Burnett, I.: Blind Speech Separation Using a Joint Model of Speech Production. IEEE Signal Processing Lett. 12 (11) (2005) 784-787
4. Zhang, Z.-L., Yi, Z.: Robust Extraction of Specific Signals with Temporal Structure. Neurocomputing 69 (7-9) (2006) 888-893
5. Barros, A.K., Cichocki, A.: Extraction of Specific Signals with Temporal Structure. Neural Computation 13 (9) (2001) 1995-2003
6. Cichocki, A., Thawonmas, R.: On-line Algorithm for Blind Signal Extraction of Arbitrarily Distributed, but Temporally Correlated Sources Using Second Order Statistics. Neural Processing Letters 12 (2000) 91-98



**Fig. 2.** Segments of the AR source signals ( $s_1, s_2, s_3$ ) and the extracted signals.  $y_1$  was extracted by the online LP-BSE algorithm in [8], while  $y_2$  was extracted by the online version of the algorithm (11).

7. Mandic, D.P., Cichocki, A.: An Online Algorithm for Blind Extraction of Sources with Different Dynamical Structures. Proc. of the 4th Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA 2003) (2003) 645-650
8. Liu, W., Mandic, D.P., Cichocki, A.: A Class of Novel Blind Source Extraction Algorithms Based on a Linear Predictor. Proc. of ISCAS 2005, pp. 3599-3602
9. Liu, W., Mandic, D.P., Cichocki, A.: Blind Second-order Source Extraction of Instantaneous Noisy Mixtures. IEEE Trans. Circuits Syst. II 53 (9) (2006) 931-935
10. Taleb, A., Cirrincione, G.: Against the Convergence of the Minor Component Analysis Neurons. IEEE Trans. Neural Networks 10 (1) (1999) 207-210
11. Feng, D.-Z., Zheng, W.-X., Jia, Y.: Neural Network Learning Algorithms for Tracking Minor Subspace in High-dimensional Data Stream. IEEE Trans. Neural Networks 16 (3) (2005) 513-521
12. Wilkinson, J.H.: The Algebraic Eigenvalue Problem. Oxford Univ. Press (1965)
13. Makeig, S., Westerfield, M., Jung, T.-P., et al.: Dynamic Brain Sources of Visual Evoked Responses. Science 295 (2002) 690-694
14. Golub, G.H., Loan, C.F.V.: Matrix Computation, 3rd ed. The John Hopkins University Press (1996)
15. Zhang, Q., Leung, Y.-W.: A Class of Learning Algorithms for Principal Component Analysis and Minor Component Analysis. IEEE Trans. Neural Networks 11 (1) (2000) 200-204
16. Chatterjee, C.: Adaptive Algorithms for First Principal Eigenvector Computation. Neural Networks 18 (2005) 145-159
17. ICALAB Toolboxes. Available: <http://www.bsp.brain.riken.jp/ICALAB>
18. Vigário, R., et al.: Independent Component Analysis for Identification of Artifacts in Magnetoencephalographic Recordings. Proc. of NIPS 1997, pp. 229-235