Proc. the 1997 International Conference on Neural Networks (ICNN'97), Houston, June, Vol. 4, pp. 2135-2140, 1997.

Dual Cascade Networks for Blind Signal Extraction

Andrzej CICHOCKI†, Ruck THAWONMAS†* and Shun-ichi AMARI‡
†Lab. for Artificial Brain Systems and ‡Lab. for Information Representation
FRP, The Institute of Chemical and Physical Research (RIKEN)
2-1 Hirosawa, Wako-shi, Saitama 351-01 JAPAN
{cia, ruck, amari}@zoo.riken.go.jp

Abstract

A new neural-network approach is presented for extracting independent source signals one-by-one from a linear mixture of them when the number of noisy mixed signals is equal to or larger than the number of sources. In this approach, two types of cascade neural networks, having similar structures, are employed. The first cascade network performs prewhitening (preprocessing) of the mixed signals by sequentially extracting principal components. From the normalized (to unit variance) prewhitened signals, the second network, then, sequentially extracts the original source signals in order according to their stochastic properties, namely, in decreasing order of absolute values of normalized kurtosis. Extensive computer simulations confirm the validity and high performance of our approach.

1. Introduction

Recovery of original signals from a linear mixture of them when the mixing coefficients are not known is called blind separation of sources. This type of problems has potential applications to many areas of science and engineering [2-4,6,7,9-14,18]. This problem can be formulated as follows. Let the sensor signals at discrete time t ($t = 0, 1, 2, \ldots$) be described by $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)$, where $\mathbf{x}(t)$ is an $n \times 1$ sensor vector, $\mathbf{s}(t)$ is an $m \times 1$ unknown source vector having independent and zero-mean elements, $\mathbf{A}$ is an $n \times m$ unknown mixing matrix, and $\mathbf{n}(t)$ is Gaussian noise. The task of the problem is to recover the m unknown sources.

Most of the algorithms in the literature employ an underlying assumption that the number of sources is known and usually equal to the number of sensors [2-4,7,10,12-14]. Thereby, those algorithms can efficiently perform separation of the source signals in a fully parallel fashion. However, in practise, the number of sources is not known and can change rapidly in time, and is usually smaller than the number of sensors, i.e., $m \leq n$ [6]. One possible approach for solving such a problem is to extract source signals sequentially (one-by-one) [5,9-11]. This method is mainly composed of two stages: one for extraction of a single source signal from the mixture and one for generation of new input mixed signals which do not contain the already extracted signals.

In general, methods for blind equalization or deconvolution problems [17] can be applied to the first stage, as done in [5,9,11,13]. Namely, extraction of an independent source signal can be achieved by maximizing (and/or minimizing) the fourth order cumulants $\kappa_4(y_1)$ subject to certain constraints, e.g., $E(y_1^2) = m_2 = 1$, or $\|\mathbf{w}_1\| = 1$, or $w_{ii} = 1$, where $y_1 \stackrel{\text{def}}{=} \mathbf{w}_1^T \mathbf{x}_1 = \sum_{j=1}^{n} w_{1j}(t) x_{1j}(t)$. For the second stage, an adaption of the orthogonal Schur eigenvalue deflation technique was used in [9]. This technique is, however, not suited for on-line, real-time applications due to its rather high complexity. In [11], the hierarchical orthogonalization technique, as used in the SGA (Stochastic Gradient Ascent) algorithm [15] and the GHA (Generalized Hebbian Algorithm) algorithm [16], was used. However, it is rather difficult to choose proper values for the coefficient constant corresponding to the orthogonalizing feedback term, unless a priori knowledge of the kurtosis of source signals is known.

In this paper, we present a neural network approach which is able to extract source signals on-line when $m \leq n$ in decreasing order according to absolute values of their normalized kurtosis. In this approach, "interesting" signals, those most deviated from Gaussian signals, are extracted first. This approach has high ap-

*The second author is supported by the Special Postdoctoral Researchers Program of RIKEN.

plicability, for instance, when the number of sources is large and only some of them are interesting or when useful signals are buried in Gaussian noises [5,10]. In addition, our approach needs no a priori knowledge of statistics of source signals. In our approach, we employ two types of simple cascade neural networks: one for performing prewhitening (or sphering) of the sensor (mixed) signals and one for extracting the source signals from the prewhitened input signals.

In the rest of the paper, we present the prewhitening cascade neural network and the extraction cascade neural network in Sections 2 and 3, respectively. We show exemplary simulation results in Section 4, and summarize by conclusions and indicating open problems in Section 5.

2. Cascade Neural Network for Prewhitening and PCA

In practise, we are often faced with ill-conditioned cases due to the fact that some specific source signals are dominant in the mixture or differences in the covariances of the source signals can be relatively large. Moreover, our optimization criteria (maximization of absolute value of normalized kurtosis) is valid under the condition that the mixed signals are prewhitened (i.e., decorrelated).

We, therefore, need to first decorrelate the sensor signals $\mathbf{x}(t)$ by a linear transformation known as prewhitening, i.e., $\mathbf{x}_1(t) = \mathbf{V}\mathbf{x}(t)$ such that $\mathbf{R}_{x_1x_1} = E\{\mathbf{x}_1(t)\mathbf{x}_1^T(t)\} = \mathbf{I}_n$.

For this purpose, we can use a simple local learning rule:

$$\mathbf{V}(t+1) = \mathbf{V}(t) + \eta(t)(\mathbf{I}_n - \mathbf{x}_1(t)\mathbf{x}_1^T(t)) \qquad (1)$$

or a global learning rule

$$\mathbf{V}(t+1) = \mathbf{V}(t) + \eta(t)(\mathbf{I}_n - \mathbf{x}_1(t)\mathbf{x}_1^T(t))\mathbf{V}(t). \qquad (2)$$

However, when $m \leq n$, these algorithms may have some problems with convergence, especially for fixed learning rate η. More complicated algorithms such as the SGA algorithm [15] and the GHA algorithm [16] also are not able to estimate precisely higher principal components corresponding to small eigenvalues. This is due to the fact that Gram-Schmidt orthogonalization is not robust with respect to numerical errors. In this section, we present a robust cascade neural network (see Fig. 1.a) which performs prewhitening of sensor signals by sequentially extracting principal components [1]. We extract principal components (PC's) sequentially by employing the concept of self-supervising

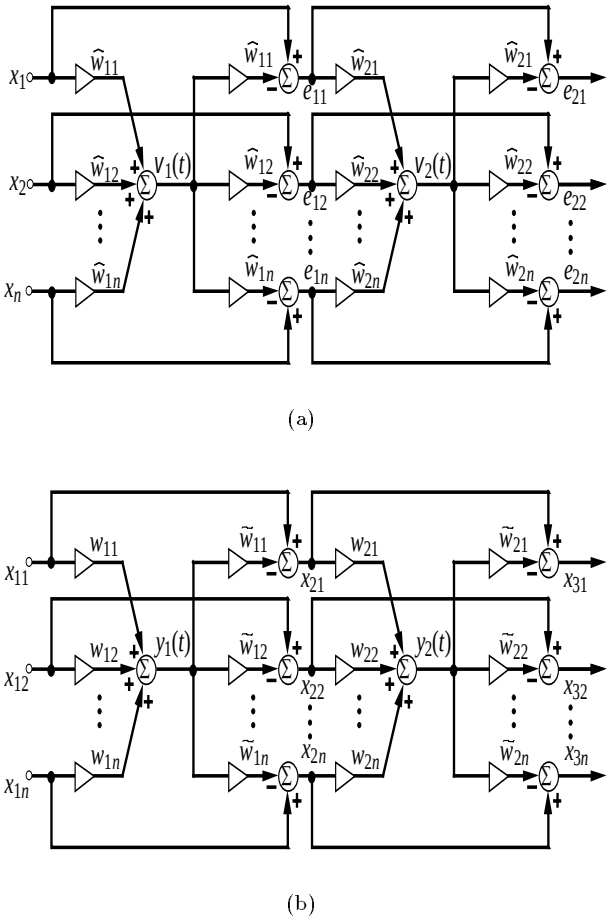


Figure 1. The architectures of cascade neural networks: (a) for PCA and prewhitening, (b) for blind extraction.

(replicator principle)) and cascade (hierarchical) neural network architecture.

Let consider a single linear neuron (see Fig. 1.a)

$$v_1(t) = \widehat{\mathbf{w}}_1^T\mathbf{x}(t) = \sum_{j=1}^{n} \widehat{w}_{1j}x_j(t), \qquad (3)$$

which would be able to extract the first principal component with eigenvalue $\lambda_1 = E\{v_1^2(t)\}$. The orthogonal vector $\widehat{\mathbf{w}}_1$ should be optimally determined in such a way that the reconstruction vector $\widehat{\mathbf{x}} = \widehat{\mathbf{w}}_1 v_1$ will reproduce (reconstruct) the input vector $\mathbf{x}(t)$ as well as possible, according to a suitable optimization criterion.

In general, the loss (cost) function can be expressed as

$$\widehat{\mathcal{J}}_1(\widehat{\mathbf{w}}_1) = \sum_{j=1}^{n} l(e_{1j}) = l(\mathbf{e}_1), \qquad (4)$$

with $\mathbf{e}_1(t) = \mathbf{x}(t) - \widehat{\mathbf{x}}(t) = \mathbf{x}(t) - \widehat{\mathbf{w}}_1 v_1(t) = [e_{11}, e_{12}, \ldots, e_{1n}]^T$, where $l(e_{1j})$ is typically a convex loss function, e.g., $l_1(e) = |e|, l_2(e) = \frac{1}{2}e^2$, or $l_3(e) = \beta \ln \cosh(e/\beta)$ [8].

The minimization of the cost function according to the standard gradient descent approach leads, after some simplifications, to the following learning rule:

$$\widehat{\mathbf{w}}_1(t+1) = \widehat{\mathbf{w}}_1(t) + \widehat{\eta}_1(t) v_1(t) \mathbf{\Psi}\big[\mathbf{x}(t) - v_1(t)\widehat{\mathbf{w}}_1(t)\big] \quad (5)$$

where $\Psi(e) = \frac{\partial l(e)}{\partial e}$, e.g., $\Psi(e) = \tanh(e/\beta)$ for $l(e) = \beta \ln \cosh(e/\beta)$.

In the special case for $l(e) = \frac{1}{2}e^2$, we obtain Oja's rule (see also [1,15]):

$$\widehat{\mathbf{w}}_1(t+1) = \widehat{\mathbf{w}}_1(t) + \widehat{\eta}_1(t) v_1(t) \mathbf{e}_1(t). \quad (6)$$

The above learning rule could be easily extended for the higher PC's using the same principle and deflation procedure. In other words, the learning rule for the extraction of the second PC $v_2(t)$ corresponding to the second largest eigenvalue $\lambda_2 = E\big[v_2^2(t)\big]$ is performed in the same way as for the first PC. However, we carry out the extraction process not directly from the input vector $\mathbf{x}(t)$ but from the error $\mathbf{e}_1(t) = \mathbf{x}(t) - \widehat{\mathbf{x}}(t) = \mathbf{x}(t) - \widehat{\mathbf{w}}_1 v_1(t)$ and $v_2(t) = \widehat{\mathbf{w}}_2^T \mathbf{e}_1(t)$ (not $v_2(t) = \widehat{\mathbf{w}}_2^T \mathbf{x}(t)$ as usually assumed in the GHA Sanger's algorithm [16]).

It can be shown that the learning algorithm for the k-th PC can be written in the general form as follows

$$\widehat{\mathbf{w}}_k(t+1) = \widehat{\mathbf{w}}_k(t) + \widehat{\eta}_k(t) v_k(t) \mathbf{\Psi}(\mathbf{e}_k(t)), \quad (7)$$

where

$$\mathbf{e}_k = \mathbf{e}_{k-1} - \widehat{\mathbf{w}}_k v_k, \quad v_k = \widehat{\mathbf{w}}_k^T \mathbf{e}_{k-1}, \quad \text{and} \quad \mathbf{e}_0(t) = \mathbf{x}(t).$$

The optimal choice of the nonlinear activation function $\Psi(e_k)$ depends on the distribution of additive noises. For Gaussian noise, the optimal one is the linear function $\Psi(e_k) = e_k, \ \forall k$.

In order to accelerate the convergence speed, we can apply RLS (recursive least squares) approach to derive an adaptive learning rule

$$\widehat{\eta}_k(t+1) = \Big[\frac{\lambda}{\widehat{\eta}_k(t)} + v_k^2(t)\Big]^{-1} \quad (8)$$

We could easily normalize the output decorrelated signals $v_k(t)$ to unit variance by scaling

$$\mathbf{x}_1(t) = \mathrm{diag}\big[\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_n^{-1/2}\big]\mathbf{v}(t), \quad (9)$$

where $\mathbf{v}(t) = [v_1(t), v_2(t), \ldots, v_n(t)]^T$, and $\lambda_k = \sqrt{\langle v_k^2 \rangle}$.

Extraction process of PC's is continued till λ_{n+1} is below a specified threshold. Using the above algorithm we could not only remove redundancy and perform whitening of sensor signals but also reduce considerably additive noise in some cases. In fact PCA divides sensor signal subspace into two subspaces: signals subspace and noise subspace. The subspace spanned by the n first PCA eigenvectors $\{\widehat{\mathbf{w}}_k\}$ is an approximation to the noiseless signal subspace.

3. Cascade Neural Network for Blind Extraction

In this section, we discuss the second type of cascade neural networks (see Fig. 1.b) which performs extraction of source signals one-by-one in decreasing order of absolute values of normalized kurtosis. The input signals to this network are the normalized prewhitened signals derived by the prewhitening network discussed in Section 2. Here we propose two associated algorithms: an extraction algorithm which extracts the designated source signals and a deflation algorithm which generates new input mixed signals containing only a mixture of signals not yet extracted.

3.1. Extraction Algorithm

To represent the stochastic properties of the source signals, we use the normalized kurtosis $\bar{\kappa}_4(y_1) = \kappa_4(y_1)/m_2^2 = E\big[y_1^4\big]/E^2\big[y_1^2\big] - 3$, rather than the standard kurtosis $\kappa_4(y_1)$. Based on the findings in [17], it can be readily shown that minimization of the loss function:

$$\mathcal{J}_1(\mathbf{w}_1) = -\frac{1}{4}|\bar{\kappa}_4(y_1)| \quad (10)$$

under the condition that mixing signals are prewhitened, i.e., decorrelated with covariance $\mathbf{R}_{x_1 x_1} = \mathbf{I}_n$: leads to a solution corresponding to one of the sources. Applying the standard gradient descent approach, we obtain a new learning rule

$$\mathbf{w}_1(t+1) = \mathbf{w}_1(t) + \eta_1(t) sgn(\bar{\kappa}_4(t)) f\big[y_1(t)\big]\mathbf{x}_1(t), \quad (11)$$

where the nonlinear function $f\big[y_1(t)\big]$ is derived by

$$f\big[y_1(t)\big] = \frac{\partial \bar{\kappa}_4(y_1)}{\partial y_1} \cong \frac{1}{m_2^2(t)}\Big[y_1^3(t) - \frac{m_4(t)}{m_2(t)} y_1(t)\Big], \quad (12)$$

and the following on-line estimations (low-pass filtering) are performed ($p = 2, 4$)

$$m_p(t+1) = (1 - \eta(t)) m_p(t) + \eta(t) y_1^p(t), \text{and} \quad (13)$$

$$\bar{\kappa}_4(t+1) = \frac{m_4(t+1)}{m_2^2(t+1)} - 3. \qquad (14)$$

The activation function $f\big[y_1(t)\big]$ (in general, $f\big[y_k(t)\big]$) is not fixed but changed during the learning process depending on the value of estimated moments $m_4(t)$ and $m_2(t)$. We confirmed experimentally that using this adaptive nonlinear function, we could recover original source signals with significantly less distortion (crosstalking) than using fixed nonlinear functions, such as that of the form $\pm(y^3 - \alpha y)$ or $(y \pm \tanh(\alpha y))$ [10,11,13], where α is a specified constant. These results were often obtained especially for source signals with positive kurtosis. We are currently conducting a study to explain this effect theoretically.

Furthermore, we add auxiliary noise to the nonlinear function, i.e., $f(\tilde{y}_1(t)) = f\big[y_1(t) + \nu_1(t)\big]$, where $\nu_1(t)$ is a Gaussian noise gradually decreasing to zero. This is done in order to avoid local minima, i.e., to ensure extraction of a source signal with the maximum absolute value of normalized kurtosis. Gaussian noises are used because they do not effect the loss function in (11); the normalized kurtosis of Gaussian noises is zero. Although we have at this moment no theoretical proof that this approach of adding noises guarantees extraction of signals in the desired order, we have found by computer experiments that the algorithm works properly. An experimental study on the effect of adding Gaussian noises or using other techniques to ensure extraction of the most interesting signal is elaborated in [18].

Finally, extraction of $y_2, \ldots, y_k$ can be performed in the same way as for y_1, using the generalized forms of the above learning rules. The corresponding input signals, however, are not the prewhitened input signals $\mathbf{x}_1$ but the deflated input signals which do not include the previously extracted signals. The algorithm for deriving those deflated signals is described in the following section.

3.2. Deflation Algorithm

Suppose that k source signals have been successfully extracted. Let y_k denote the last extracted signal. We now generate the new input vector $\mathbf{x}_{k+1}$ which will not include the already extracted signals $(y_1, \ldots, y_k)$ by the linear transformation (see Fig. 1.b)

$$\mathbf{x}_{k+1}(t) \stackrel{\text{def}}{=} \mathbf{x}_k(t) - \widetilde{\mathbf{w}}_k(t) y_k(t). \qquad (15)$$

Note that the input vector $\mathbf{x}_k(t)$ was previously derived such that it does not include $(y_1, \ldots, y_{k-1})$. The goal of the above transformation is to minimize the loss function (generalized energy)

$$\widetilde{\mathcal{J}}_k(\widetilde{\mathbf{w}}_k) = \rho(\mathbf{x}_{k+1}) = \sum\nolimits_{j=1}^{n} \rho(x_{k+1,j}), \qquad (16)$$

where $y_k = \mathbf{w}_k^T \mathbf{x}_k$, $\mathbf{x}_k = \big[x_{k,1}, x_{k,2}, \ldots, x_{k,n}\big]^T$, $\rho(\mathbf{x}_k)$ is a loss function, e.g., $\rho(\mathbf{x}_k) = \frac{1}{2}\|\mathbf{x}_k\|^2$, with $\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta_k(t) sgn(\bar{\kappa}_4(t)) f\big[\tilde{y}_k(t)\big]\mathbf{x}_k(t)$.

Applying the standard gradient descent approach, we then derive the following rule

$$\widetilde{\mathbf{w}}_k(t+1) = \widetilde{\mathbf{w}}_k(t) + \tilde{\eta}_k(t) y_k(t) g(\mathbf{x}_{k+1}(t)), \qquad (17)$$

where $g(\mathbf{x}_{k+1}) = \big[g(x_{k+1,1}), \ldots, g(x_{k+1,n})\big]^T$ and $g(x_{k+1,j}) \stackrel{\text{def}}{=} \frac{\partial \rho(\mathbf{x}_{k+1})}{\partial x_{k+1,j}}$, e.g., $g(x_{k+1,j}) = x_{k+1,j}$ for $\rho(x_{k+1,j}) = \frac{1}{2} x_{k+1,j}^2$.

From the loss function in use, it can be readily shown that if all source signals have been extracted, every element in the new deflated input vector will converge to zero. Hence, we can terminate adding of a new processing unit for extraction of the next signal y_{k+1} if the amplitude of each element in the new deflated input vector $\mathbf{x}_{k+1}$ is less than a given threshold, i.e., $|x_{i,k+1}| < \sigma \quad \forall i$.

4. Computer simulations

We confirmed the validity and performance of our approach using extensive computer simulations for a variety of problems. We initialized all the weights such that they had random values in the range -0.1 and 0.1. We set the initial learning rates in the extraction network for extraction $\eta_k(t)$ and deflation $\tilde{\eta}_k(t)$ to 0.05. To these learning rates, we applied the learning of learning rate scheme as discussed in [4,5]. For extraction, we added the following noise $\nu_k(t)$ to the nonlinear function $f(y_k(t))$: $\nu_k(t) = n_k(t)$ for $t \leq 500$; 0 otherwise, where $n_k(t)$ is a Gaussian noise with mean 0.0 and variance 1.0. In both cascade networks, execution of a next cascade is delayed for 5000 thousand time steps after initiating execution of the previous cascade. Below, due to limit of space, we only present an illustrative example of typical results.

Fig. 2 shows the results of extraction of three unknown signals (Fig. 2.a) from a mixture of them received at four sensors (Fig. 2.b), where $\bar{\kappa}_4(s_1) = -1.2$, $\bar{\kappa}_4(s_2) = -1.5$, and $\bar{\kappa}_4(s_3) = 0.49$. The prewhitened signals, obtained by the prewhitening network, are shown in Fig. 2.c. Here, we used the first three signals (principal components) because the variance of the fourth principal component was extremely low in comparison to those of the previous principal components ($\lambda_1 = 0.42, \lambda_2 = 0.24, \lambda_3 = 0.04$, and

$\lambda_4 = 4 \times 10^{-8}$). The extracted source signals, obtained by the extraction network, are shown in Fig. 2.d, where y_k stands for the k-th extracted signal. By visual comparison of Fig. 2.a and 2.d, the source signals were successfully extracted, and, in addition, in decreasing order of absolute values of normalized kurtosis. Finally, the deflated signals are shown in Fig. 2.e. Having the amplitude of every element of $\mathbf{x}_4$ relatively very small confirmed that the number of active sources in the mixed signals, which was not known to the system, was three.

5. Conclusions

We have presented a dual cascade neural-network approach for blind signal extraction. The first cascade neural network, which performs robust prewhitening of mixed signals in a sequential fashion based on the concept of self-supervising, allows us to cope with practical cases where the mixed signals are ill-conditioned and/or noisy or where the number of mixed signals is equal to or larger than the number of sources. The second cascade neural network performs sequential extraction of source signals from a mixture of them in decreasing order of the degrees of being interesting. Namely, the most interesting (most deviated from Gaussian signals) signal is extracted first, then the second interesting signal is extracted next, and so on. The developed learning algorithms are purely local and are biologically plausible; they could be considered as a generalization or extension of Hebbian/anti-Hebbian rules. The proposed methodology can be extended to complex-valued signals and mixtures as well as multi-channel blind signal deconvolution.

References

[1] S. Amari "Neural theory of association and concept-formulation", *Biological Cybernetics.*, Vol. 26, 1977, pp. 175-185.

[2] S. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind signal separation", *NIPS 95*, MIT Press, Vol. 8, 1996, pp. 757-763.

[3] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation", *IEEE Trans. on Signal Processing*, Vol. 44, No. 12, Dec 1996, pp. 3017-3030.

[4] A. Cichocki, S. Amari, M. Adachi, and W. Kasprzak, "Self-adaptive neural networks for blind separation of sources", *ISCAS 96*, May 1996, Vol. II, pp. 157-160.

[5] A. Cichocki, S. Amari and R. Thawonmas, "Blind signal extraction using self adaptive non-linear hebbian learning rule", *NOLTA'96*, October 1996, pp. 377-380.

[6] A. Cichocki, W. Kasprzak and S. Amari, "Neural network approach to blind separation of enhancement of images", *EUSIPCO 96*, September 1996, Vol. I, pp. 579-582.

[7] A. Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources", *IEEE Trans. Circuits and Systems-I*, Vol. 43, Nov. 1996, pp. 894-906.

[8] A. Cichocki and R. Unbehauen, "Robust estimation of principal components by using neural network learning algorithms", *Electronics Letters*, Vol. 29, No. 21, 1993, pp. 1869-1870.

[9] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A deflation approach", *Signal Processing*, Vol. 45, 1995, pp. 59-83.

[10] M. Girolami and C. Fyfe, "Blind separation of sources using exploratory projection pursuit networks", *Int. Conf. on Eng. Applications of Neural Networks*, 1996, pp. 249-252.

[11] A. Hyvärinen and E. Oja, "A neuron that learns to separate one signal from a mixture of independent sources", *ICNN 96*, June 1996, Vol. I, pp. 62-67.

[12] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture", *Signal Processing*, Vol. 24, 1991, pp. 1-20.

[13] Z. Malouche and O. Macchi, "Extended anti-hebbian adaptation for unsupervised source extraction", *ICASSP 96*, May 1996, pp. 1664-1667.

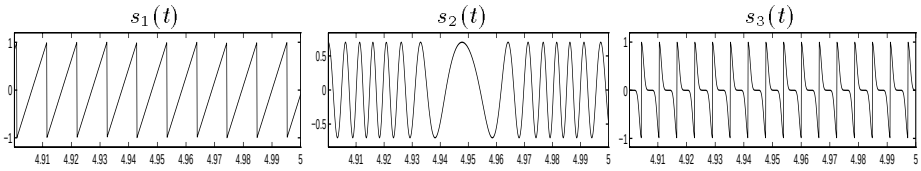
[14] E. Oja and J. Karhunen, "Signal separation by nonlinear hebbian", *Computational Intelligence -a Dynamic System Perspective*, pp. 83-97, IEEE Press, New York, 1995.

[15] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix", *Journal of Math. Analysis and Applications 106*, pp. 69-84, 1985.

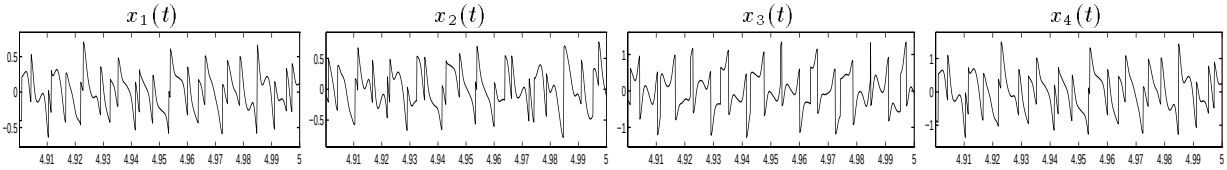
[16] T. Sanger, "Optimal unsupervised learning in a single-layered linear feedforward network", *Neural Networks*, Vol. 2, 1989, pp. 459-473.

[17] O. Shalvi and E. Weinstein "Universal method for blind deconvolution", *Blind Deconvolution*, S. Haykin, Ed., pp. 121-180, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994.

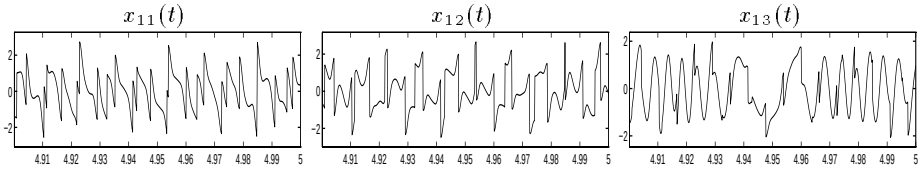
[18] R. Thawonmas and A. Cichocki "Blind extraction of source signals with specified stochastic features", *ICASSP 97*, April 1997, Vol. 4, pp. 3353-3356.



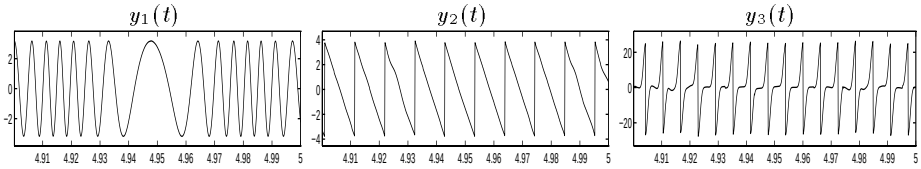
(a) **original source signals**



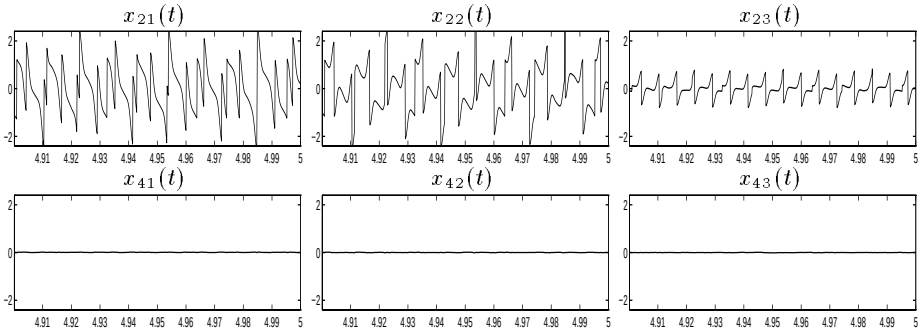
(b) **mixed signals**



(c) **prewhitened signals**



(d) **extracted signals**



(e) **deflated signals after the 1-th, and 3-rd processing unit, respectively**

Figure 2. A typical result of extraction of three sources received at four sensors (with 10 KHz sampling rate).