

# FLEXIBLE INDEPENDENT COMPONENT ANALYSIS

Seungjin CHOI<sup>†</sup>, Andrzej CICHOCKI<sup>††</sup>, Shunichi AMARI<sup>††</sup>

<sup>†</sup>School of Electrical and Electronics Engineering  
Chungbuk National University  
48 Kaeshin-dong, Cheongju  
Chungbuk 361-763, KOREA  
Email: schoi@engine.chungbuk.ac.kr

<sup>††</sup>Brain-Style Information Systems Research Group  
Brain Science Institute, RIKEN  
2-1 Hirosawa, Wako-shi  
Saitama 351-01, JAPAN  
Email: {cia,amari}@brain.riken.go.jp

**Abstract.** We present a flexible independent component analysis (ICA) algorithm which can separate mixtures of sub- and super-Gaussian source signals with self-adaptive nonlinearities. The flexible ICA algorithm in the framework of natural Riemannian gradient, is derived using the parameterized generalized Gaussian density model. The nonlinear function in the flexible ICA algorithm is self-adaptive and is controlled by Gaussian exponent. Computer simulation results confirm the validity and high performance of the proposed algorithm.

## INDEPEDENT COMPONENT ANALYSIS

Independent component analysis is a fundamental statistical method encountered in many applications such as feature extraction, digital communications, robust speech recognition, image processing, and biomedical signal analysis (like ECG, EEG, and MEG). In many applications, the sensory signals (observations obtained from multiple sensors) are generated by an linear generative model which is unknown to us. In other words, observations are linear instantaneous mixtures of unknown source signals. It is desirable to recover the source signals from observations by building the recognition model.

Let us assume that the  $m$  dimensional vector of observed signals,  $\mathbf{x}(t) = [x_1(t), \dots, x_m(t)]^T$  is generated by an unknown linear generative model,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad (1)$$

where  $\mathbf{s}(t) = [s_1(t), \dots, s_n(t)]^T$  is the  $n$  dimensional vector whose elements are called sources. The matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is called a mixing matrix. It is assumed that source signals  $\{s_i(t)\}$  are mutually independent. The number of sensors,  $m$  is greater than or equal to the number of sources,  $n$ .

The task of ICA is to recover source signals  $\mathbf{s}(t)$  from the observations  $\mathbf{x}(t)$  without the knowledge of  $\mathbf{A}$  nor  $\mathbf{s}(t)$ . This is often called as blind source separation (BSS). We build a recognition model which transforms the observations  $\mathbf{x}(t)$  to the network output signals  $\mathbf{y}(t)$  whose elements are statistically mutually independent, so that the output signals  $\mathbf{y}(t)$  are possibly scaled estimates of source signals  $\mathbf{s}(t)$ . Inherently there are two indeterminacies in ICA [1]: (1) scaling ambiguity; (2) permutation ambiguity. That is, the recovered signals  $\mathbf{y}(t)$  by a recognition model are  $\mathbf{y}(t) = \mathbf{P}\mathbf{A}\mathbf{s}(t)$ , where  $\mathbf{P}$  is a permutation matrix and  $\mathbf{A}$  is some nonsingular diagonal matrix.

Since Jutten and Herault [2] proposed a linear feedback network with a simple unsupervised learning algorithm, several methods have been developed. Cichocki *et al.* [3, 4] proposed a robust, flexible algorithm with equivariant properties. Comon [1] gave a good insight to ICA problem from the statistical point of view. Bell and Sejnowski [5] adopted an information maximization principle to find a solution to ICA problem. Maximum likelihood estimation [6] was proposed by Pham *et al.* and was elaborated in [7, 8]. The nonlinear extension of PCA was extensively studied in [9, 10]. Serial updating rule was introduced by Cardoso and Laheld [11] and the resulting algorithm was shown to have equivariant performance. Independently, natural gradient was proposed and applied to ICA by Amari *et al.* [12]. Conditions on cross-cumulants for the separation of source signals were investigated in [13–15].

## Natural Riemannian Gradient

Let us consider a linear feedforward memoryless neural network which maps the observation  $\mathbf{x}(t)$  to  $\mathbf{y}(t)$

$$\mathbf{y}(t) = \mathbf{W}(t)\mathbf{x}(t), \quad (2)$$

where  $(i, j)$ th element of the matrix  $\mathbf{W}(t)$ , i.e.,  $w_{ij}(t)$  represents a synaptic weight between  $y_i(t)$  and  $x_j(t)$ . In the limit of zero noise, for the square ICA problem (equal number of sources and sensors, the result can be easily extended to the case  $m > n$ ), maximum likelihood or mutual information minimization suggest the following loss function [12]:

$$L(\mathbf{W}(t)) = -\log |\det \mathbf{W}(t)| - \sum_{i=1}^n \log p_i(y_i(t)), \quad (3)$$

where  $p_i(\cdot)$  represent the probability density function. Let us define

$$f_i(y_i(t)) = -\frac{d \log p_i(y_i(t))}{dy_i(t)}. \quad (4)$$

With this definition, the gradient of the loss function (3) is

$$\begin{aligned}\nabla L(\mathbf{W}(t)) &= \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}(t)} \\ &= -\mathbf{W}^{-T}(t) + \mathbf{f}(\mathbf{y}(t))\mathbf{x}^T(t),\end{aligned}\quad (5)$$

where  $\mathbf{f}(\mathbf{y}(t))$  is the element-wise function whose  $i$ th component is  $f_i(y_i(t))$ .

The natural Riemannian gradient (denoted by  $\tilde{\nabla}L(\mathbf{W}(t))$ ) learning algorithm for  $\mathbf{W}(t)$  is given by [4, 11, 16]

$$\begin{aligned}\mathbf{W}(t+1) &= \mathbf{W}(t) - \eta_t \tilde{\nabla}L(\mathbf{W}(t)) \\ &= \mathbf{W}(t) - \eta_t \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}(t)} \mathbf{W}^T(t) \mathbf{W}(t) \\ &= \mathbf{W}(t) + \eta_t \{\mathbf{I} - \mathbf{f}(\mathbf{y}(t))\mathbf{y}^T(t)\} \mathbf{W}(t).\end{aligned}\quad (6)$$

### Natural Riemannian Gradient in Orthogonality Constraint

Natural Riemannian gradient in orthogonality constraint has been recently proposed by Amari [17]. Let us assume that the observation vector  $\mathbf{x}(t)$  has already been whitened by preprocessing and source signals are normalized, i.e.,

$$\begin{aligned}E\{\mathbf{x}(t)\mathbf{x}^T(t)\} &= \mathbf{I}_m, \\ E\{\mathbf{s}(t)\mathbf{s}^T(t)\} &= \mathbf{I}_n.\end{aligned}\quad (7)$$

From (7) and (8), we have

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}_m. \quad (9)$$

The  $m$  row vectors of  $\mathbf{A}$  are orthogonal  $n$  dimensional unit vectors. The set of  $n$  dimensional subspaces in  $\mathbb{R}^m$  is called Stiefel manifold. The natural Riemannian gradient in the Stiefel manifold was calculated by Amari [17]

$$\tilde{\nabla}L(\mathbf{W}(t)) = \nabla L(\mathbf{W}(t)) - \mathbf{W}(t)\{\nabla L(\mathbf{W}(t))\}^T \mathbf{W}(t). \quad (10)$$

Using this result, the natural gradient is given by

$$\tilde{\nabla}L(\mathbf{W}(t)) = \mathbf{f}(\mathbf{y}(t))\mathbf{x}^T(t) - \mathbf{y}(t)\mathbf{f}^T(\mathbf{y}(t))\mathbf{W}(t). \quad (11)$$

Then the learning algorithm for  $\mathbf{W}(t)$  is given by

$$\begin{aligned}\mathbf{W}(t+1) &= \mathbf{W}(t) - \eta_t \tilde{\nabla}L(\mathbf{W}(t)) \\ &= \mathbf{W}(t) - \eta_t \{\mathbf{f}(\mathbf{y}(t))\mathbf{x}^T(t) - \mathbf{y}(t)\mathbf{f}^T(\mathbf{y}(t))\mathbf{W}(t)\}.\end{aligned}\quad (12)$$

It should be noted that when  $m = n$ ,  $\mathbf{A}$  or  $\mathbf{W}(t)$  is orthogonal, this reduces to the Cardoso and Laheld's algorithm [11] described as

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta_t \{\mathbf{f}(\mathbf{y}(t))\mathbf{y}^T(t) - \mathbf{y}(t)\mathbf{f}^T(\mathbf{y}(t))\} \mathbf{W}(t). \quad (13)$$

## GENERALIZED GAUSSIAN DENSITY MODEL FOR SOURCES

Optimal nonlinear activation function  $f_i(y_i(t))$  is calculated by (4). However, it requires the knowledge of the probability distribution of source signals which are not available to us. A variety of hypothesized density model has been used. For example, for super-Gaussian source signals, unimodal or hyperbolic-Cauchy distribution model [7] leads to the nonlinear function given by

$$f_i(y_i(t)) = \tanh(\beta y_i(t)). \quad (14)$$

Such sigmodal function was also used in [5]. For sub-Gaussian source signals, cubic nonlinear function  $f_i(y_i(t)) = y_i^3(t)$  has been a favorite choice. For mixtures of sub- and super-Gaussian source signals, according to the estimated kurtosis of the extracted signals, nonlinear function can be selected from two different choices [18]. (for example, either  $f_i(y_i(t)) = \tanh(\beta y_i(t))$  or  $f_i(y_i(t)) = y_i^3(t)$ ) Several approaches [19, 20] are already available.

This paper present a flexible nonlinear function derived using generalized Gaussian density model. It will be shown that the nonlinear function is self-adaptive and controlled by Gaussian exponent. It is not a form of fixed nonlinear function (see also [20, 21])

### The Generalized Gaussian Distribution

The *generalized Gaussian* probability distribution is a set of distributions parameterized by a positive real number  $\alpha$ , which is usually referred to as the *Gaussian exponent* of the distribution. The Gaussian exponent  $\alpha$  controls the “peakiness” of the distribution. The probability density function (PDF) for a generalized Gaussian is described by

$$p(y; \alpha) = \frac{\alpha}{2\sigma\Gamma(\frac{1}{\alpha})} e^{-|\frac{y}{\sigma}|^\alpha}, \quad (15)$$

where  $\Gamma(x)$  is Gamma function given by

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt. \quad (16)$$

Note that if  $\alpha = 1$ , the distribution becomes the standard “Laplacian” distribution. If  $\alpha = 2$ , the distribution is standard normal distribution.

### The Moments of the Generalized Gaussian Distribution

In order to fully understand the generalized Gaussian distribution, it is useful to look at its moments (specially 2nd and 4th moments which give the kurtosis). The  $n$ th moment of the generalized Gaussian distribution is given

by

$$M_n = \int_{-\infty}^{\infty} y^n p(y; \alpha) dy. \quad (17)$$

If  $n$  is odd, the integrand is the product of an even function and an odd function over the whole real line, which integrates to zero. In particular, this implies that the mean of the distribution given in (15) is zero and it is symmetric about its mean (which means its skewness is zero).

The even moments, on the other hand, completely characterize the distribution. In computing these moments, we use the following integral formula (see pp. 386 in [22])

$$\int_0^{\infty} y^{\nu-1} e^{-\mu y^a} dy = \frac{1}{a} \mu^{-\frac{1}{a}} \Gamma\left(\frac{\nu}{a}\right). \quad (18)$$

The 2nd moment of the generalized Gaussian distribution is determined by

$$\begin{aligned} M_2 &= \int_{-\infty}^{\infty} y^2 p(y; \alpha) dy \\ &= 2 \int_0^{\infty} y^2 \frac{\alpha}{2\sigma\Gamma(\frac{1}{\alpha})} e^{-|\frac{y}{\sigma}|^\alpha} dy. \end{aligned} \quad (19)$$

We are integrating only over the positive values of  $y$ , we can remove the absolute value in the exponent. Thus

$$M_2 = \frac{\alpha}{\sigma\Gamma(\frac{1}{\alpha})} \int_0^{\infty} y^2 e^{-(\frac{y}{\sigma})^\alpha} dy. \quad (20)$$

Making the substitution  $z = \frac{y}{\sigma}$  ( $dy = \sigma dz$ ), we find

$$M_2 = \frac{\alpha\sigma^2}{\Gamma(\frac{1}{\alpha})} \int_0^{\infty} z^2 e^{-z^\alpha} dz. \quad (21)$$

Invoking the integral formula (18), we have

$$M_2 = \sigma^2 \frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}. \quad (22)$$

In similar way, we can find the 4th moment given by

$$M_4 = \sigma^4 \frac{\Gamma(\frac{5}{\alpha})}{\Gamma(\frac{1}{\alpha})}. \quad (23)$$

In general, the  $(2k)$ th moment is given by

$$M_{2k} = \sigma^{2k} \frac{\Gamma(\frac{2k+1}{\alpha})}{\Gamma(\frac{1}{\alpha})}. \quad (24)$$

## Kurtosis and Gaussian Exponent

The kurtosis is a nondimensional quantity. It measures the relative peakedness or flatness of a distribution. A distribution with positive kurtosis is termed *leptokurtic* (super-Gaussian). A distribution with negative kurtosis is termed *platykurtic* (sub-Gaussian). The kurtosis of the distribution is defined in terms of the 2nd- and 4th-order moments as

$$\kappa(y) = \frac{M_4}{M_2^2} - 3, \quad (25)$$

where the constant term  $-3$  makes the value zero for standard normal distribution.

For a generalized Gaussian distribution, the kurtosis can be expressed in terms of the Gaussian exponent, given by

$$\kappa_\alpha = \frac{\Gamma(\frac{5}{\alpha})\Gamma(\frac{1}{\alpha})}{\Gamma^2(\frac{3}{\alpha})} - 3. \quad (26)$$

The plot of kurtosis  $\kappa_\alpha$  versus the Gaussian exponent  $\alpha$  for leptokurtic and platykurtic signals are shown in Figure 1 and 2, respectively.

## THE FLEXIBLE ICA ALGORITHM

From the parameterized generalized Gaussian density model, the nonlinear function in the algorithms (6) and (13) is given by

$$f_i(y_i(k)) = |y_i(k)|^{\alpha-1} \text{sign}(y_i(k)). \quad (27)$$

Note that for  $\alpha = 2$ , (27) becomes a linear function which can be derived from the Gaussian density model for sources, for  $\alpha = 1$ , (27) becomes a sign function (which can be derived from the Laplacian density model for sources), for  $\alpha = 4$ , (27) becomes a cubic function which is known to be a good choice for sub-Gaussian signals. For uniform distributed source signals, the ideal optimal choice of  $\alpha$  will be a infinity, however, it will cause a numerical problem. From Figure 2, we can observe that the kurtosis does not change much according to the Gaussian exponent  $\alpha$ . This explains why a cubic function  $f_i(y_i(k)) = |y_i(k)|^3 \text{sign}(y_i(k))$  works well for the mixture of sub-Gaussian signals. We can choose  $\alpha = 6, 8$  or an ever bigger even number, but these choices do not help the separation. An interesting point is the super-Gaussian case. We can see a reasonably large change of kurtosis with respect to the change of the Gaussian exponent  $\alpha$  from Figure 1. For spike signals (with high kurtosis), a good choice of  $\alpha$  is close to one or even less than one. When we choose a tanh function, i.e.,  $f_i(y_i(k)) = \tanh(\beta y_i(k))$ , we know that the value of  $\beta$  should be a reasonably big number, say 10 for the separation of high kurtosis signals. This can be explained from Figure 1. In practice, we can monitor the estimates of kurtosis of extracted signals, then we can make a good selection of  $\alpha$ . From our experience, a small change of  $\alpha$  has no effect on the performance of the algorithm given in (6) and (13).

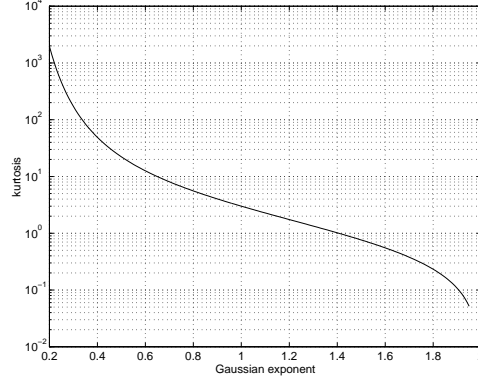


Figure 1: The plot of kurtosis  $\kappa_\alpha$  versus Gaussian exponent  $\alpha$  for leptokurtic distribution

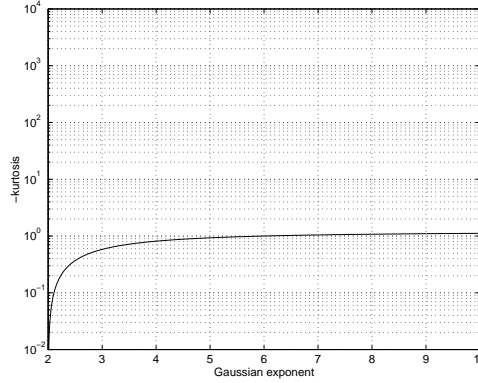


Figure 2: The plot of absolute value of kurtosis  $\kappa_\alpha$  versus Gaussian exponent  $\alpha$  for platykurtic distribution

## COMPUTER SIMULATION RESULTS

The observation vector  $\mathbf{x}(t)$  was generated from the instantaneous mixtures of two super-Gaussian source signals and one sub-Gaussian source signal through a mixing matrix  $\mathbf{A}$  which is given by

$$A = \begin{bmatrix} 0.1549 & 0.1405 & 0.3916 \\ 0.5258 & 0.2041 & 0.9370 \\ 0.2047 & 0.5108 & 0.4310 \end{bmatrix}. \quad (28)$$

Three source signals are shown in Figure 3. The kurtosis of each source signal is .41, 138.84, -1.2, respectively. Three different Gaussian exponent  $\alpha$  were used: (1)  $\alpha = .8$  when the estimated kurtosis of recovered signal  $y_i(t)$  is greater than 20; (2)  $\alpha = 1$  when the estimated kurtosis of recovered signal is between 0 and 20; (3)  $\alpha = 4$  when the estimated kurtosis of recovered signal is negative. Three mixture signals  $\mathbf{x}(t)$  are shown in Figure 4. The constant learning rate  $\eta_t = .001$  was used. The recovered signals  $\mathbf{y}(t)$  are plotted in

Figure 5. It can be observed that after 3000 iterations, source signals are well separated.

## CONCLUSION

We have present the flexible ICA algorithm (in the framework of natural Riemannian gradient) where the self-adaptive nonlinear function was derived using generalized Gaussian density model for the probability distributions of source signals. We have shown that flexible ICA algorithm can separate the mixtures of sub- and super-Gaussian signals with self-adaptive nonlinearities which is controlled by Gaussian exponent.

## REFERENCES

- [1] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
- [2] C. Jutten and J. Herault, "Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1–10, 1991.
- [3] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert, "A new on-line adaptive learning algorithm for blind separation of source signals," in *International Joint Conference on Neural Networks*, 1994, pp. 406–411.
- [4] 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: Fundamental Theory and Applications*, vol. 43, pp. 894–906, 1996.
- [5] A. Bell and T. Sejnowski, "An information maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.
- [6] D. T. Pham, P. Garrat, and C. Jutten, "Separation of mixture of independent sources through a maximum likelihood approach," in *European Signal Processing Conference*, 1992, pp. 771–774.
- [7] D. J. C. MacKay, "Maximum likelihood and covariant algorithms for independent component analysis," 1996, University of Cambridge, Cavendish Laboratory, Draft 3.7.
- [8] B. Pearlmutter and L. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., 1997, pp. 613–619.

- [9] J. Karhunen, "Neural approaches to independent component analysis," in *European Symposium on Artificial Neural Networks*, 1996, pp. 249–266.
- [10] E. Oja, "The nonlinear PCA learning rule and signal separation - mathematical analysis," 1995.
- [11] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, 1996.
- [12] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. 1996, vol. 8, pp. 757–763, MIT press.
- [13] J. P. Nadal and N. Parga, "Redundancy reduction and independent component analysis: Conditions on cumulants and adaptive approaches," *Neural Computation*, vol. 9, pp. 1421–1456, 1997.
- [14] S. Choi, R. Liu, and A. Cichocki, "A spurious equilibria-free learning algorithm for the blind separation of non-zero skewness signals," *Neural Processing Letters*, vol. 7, pp. 61–68, 1998.
- [15] S. Choi and A. Cichocki, "A linear feedforward neural network with lateral feedback connections for blind source separation," in *IEEE Signal Processing Workshop on Higher-order Statistics*, Banff, Canada, 1997, pp. 349–353.
- [16] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, 1998.
- [17] S. Amari, "Natural gradient for over- and under-complete bases in ICA," 1998, submitted to *Neural Computation*.
- [18] S. C. Douglas, A. Cichocki, and S. Amari, "Multichannel blind separation and deconvolution of sources with arbitrary distributions," in *Neural Networks for Signal Processing 7*, J. Principe, L. Gile, N. Morgan, and E. Wilson, Eds., 1997, pp. 436–445.
- [19] M. Girolami and C. Fyfe, "Generalized independent component analysis through unsupervised learning with emergent bussgang properties," in *IEEE International Conference on Neural Networks*, 1997, pp. 1788–1791.
- [20] A. Cichocki, I. Sabala, S. Choi, B. Orsier, and R. Szupiluk, "Self-adaptive independent component analysis for sub-gaussian and super-gaussian mixtures with unknown number of source signals," in *International Symposium on Nonlinear Theory and Applications*, 1997, pp. 731–734.

- [21] A. Cichocki, S. Douglas, S. Amari, and P. Mierzejewski, "Independent component analysis for noisy data," in *Proc. of Int. Workshop on Independence and Artificial Neural Networks*, Tenerife, Spain, 1998, pp. 52–58.
- [22] I. S. Gradshteyn, I. M. Ryzhik, and A. Jeffrey, *Table of Integrals, Series, and Products*, Academic Press, 1994.

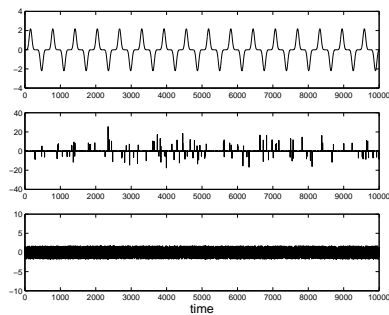


Figure 3: Three original source signals, from top to bottom,  $s_1, s_2, s_3$ .

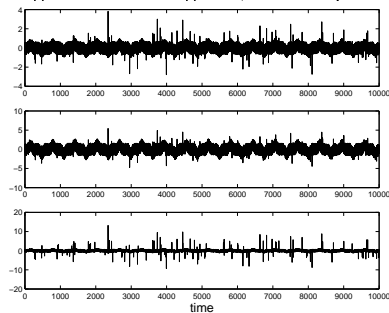


Figure 4: Three mixtures, from top to bottom,  $x_1, x_2, x_3$ .

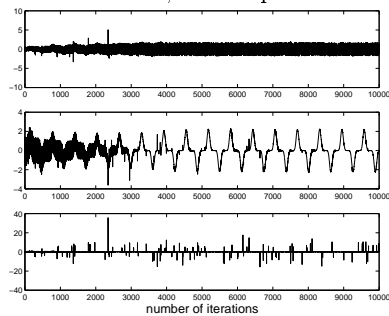


Figure 5: Three recovered signals, from top to bottom,  $y_1, y_2, y_3$ .